SCALABLE VIDEO COMMUNICATIONS: BITSTREAM EXTRACTION ALGORITHMS FOR STREAMING, CONFERENCING AND 3DTV

A Thesis
Presented to
The Academic Faculty

by

Ramanathan Palaniappan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

SCALABLE VIDEO COMMUNICATIONS: BITSTREAM EXTRACTION ALGORITHMS FOR STREAMING, CONFERENCING AND 3DTV

Approved by:

Professor Nikil Jayant, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Christopher F. Barnes
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Constantine Dovrolis
College of Computing
Georgia Institute of Technology

Professor Ghassan Al-Regib
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 10 August 2011

To Amma, Appa,

Udayavar, Nammazhwar,

and the Divya Dhampathis of Srirangam.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my advisor, Prof. Nikil Jayant for his support and faith in my abilities throughout my graduate studies at Georgia Tech. This doctoral dissertation would not have been possible without his step-by-step guidance. I have always admired the depth of his technical knowledge, meticulousness and attention to details. I would like to express my gratitude towards him for providing me with all the facilities needed to complete my doctoral research successfully and for giving me an opportunity to explore and further my knowledge in the field. I consider it as a great privilege to have worked under him, as one of his PhD students. Next, I would like to thank all the members of my thesis committee, Prof. Raghupathy Sivakumar, Prof. Ghassan Al-Regib, Prof. Christopher Barnes and Prof. Constantine Dovrolis for taking the time to read my thesis, attend my defense and offer insightful comments. Their feedback was very useful and helped in improving the thesis.

I would like to express my appreciation to all the staff members at GCATT. I am extremely grateful to Barbara Satterfield for scheduling all my meetings with Dr. Jayant over the past five years in an organized fashion. A special thanks to her for taking care of the paperwork and the scheduling process for both my PhD proposal and defense. Her coordination and support made the entire process a very pleasant experience for everyone involved. Next, I would like to thank Rex Smith for providing technical assistance with all our projects. His organization of the Las Vegas trip for our participation in the NABSHOW, 2011 deserves a special mention. His complete guidance from start to end made the entire show a very smooth ride. A big thanks to Tina Clonts for taking care of all the monetary issues involved over the past five years. She has always been there when it comes to reimbursements, tuition waivers,

equipment purchases made for our lab, etc. Finally, many thanks to JoAnna Shorter for arranging all my travel to conferences. From booking my flight tickets to making hotel arrangements, she has been the reason why our trips have always been such a wonderful experience.

My stay at Georgia Tech has been made memorable by all my friends. I would like to express my heartfelt thanks to all of them for always being there for me and cheering me up during difficult times. I have thoroughly enjoyed the technical conversations that I have had with my labmates Nitin Suresh, Eun Seok Ryu, Jeannie Lee, Sourabh Khire and Saunya Williams. I have been fortunate to have been surrounded by a group of excellent friends. Many thanks to Gokul Kumar, Ramanan Subramanian, Arunkumar Subramanian, Girish Venkatesh, Lavanavarjit Ragavan, Karthik Ramakrishnan, Karthekeyan Chandrasekaran, Mahesh Viswanathan for making a big difference in my life. Those long discussions about Srivaishnavam with Ramanan, carnatic music technicalities with Gokul and hour-long gossips with Arun will always stay green in my memory.

Finally, many thanks to my parents, my sister Meena and brother-in-law Nagappan, their kids Sriram and Srinidhi for always believing in me and standing by me patiently all my life. Words are not sufficient to express my gratitude towards them. I am indebted to them for all the sacrifices they have made so that I can pursue this PhD degree. They made all the effort worth it. A big thanks to my mentor Sri U. Ve. Velukkudi Krishnan Swamy whose spiritual discourses have provided me with a constant source of encouragement and transformed my life. Last, but not least, I thank our Poorvacharyas, Azhwars, Piratti and Perumal for showering their unlimited divine grace upon me throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This research investigates scalable video communications and its applications to video streaming, conferencing and 3DTV. Scalable video coding (SVC) is a layer-based encoding scheme that provides spatial, temporal and quality scalability. Heterogeneity of the Internet and clients' operating environment necessitate the adaptation of media content to ensure a satisfactory multimedia experience. SVC's layer structure allows the extraction of partial bitstreams at reduced spatial, quality and temporal resolutions that adjust the media bitrate at a fine granularity to changes in network state. The main focus of this research work is in developing such extraction algorithms in the context of SVC. Based on a combination of metadata computations and prediction mechanisms, these algorithms evaluate the quality contribution of each layer in the SVC bitstream and make extraction decisions that are aimed at maximizing video quality while operating within the available bandwidth resources. These techniques are applied in two-way interaction and one-way streaming of 2D and 3D content. Depending on the delay tolerance of these applications, rate-distortion optimized extraction algorithms are proposed. For conferencing applications, the extraction decisions are made over single frames and frame pairs due to tight end-to-end delay constraints. The proposed extraction algorithms for 3D content streaming maximize the overall perceived 3D quality based on human stereoscopic perception. When compared to current extraction methods, the new algorithms offer better video quality at a given bitrate while performing lesser number of metadata computations in the post-encoding phase. The solutions proposed for each application achieve the recurring goal of maintaining the best possible level of end-user quality of multimedia experience in spite of network impairments.

CHAPTER I

INTRODUCTION

Video streaming and conferencing are two major forms of multimedia communication. Such communication over the Internet has seen an unprecedented growth in the past decade. TV broadcasts and services such as VOD (video-on-demand) over the Internet have become enormously popular. With the introduction of affordable 3DTVs in the consumer market combined with the release of 3D movies, there is a fresh interest in streaming of such content too. Recently, Comcast streamed the Masters Tournament live in 3D to its customers. A number of factors have contributed to this success including advanced multimedia technologies, improved backbone network infrastructure, affordable broadband connectivity, etc. Rapid improvements in mobile networks and portable device technology such as netbooks and smart phones have led to a heavy usage of mobile Internet. This has added another dimension to media streaming since these devices vary widely in their processing power, display size and their network connectivities. Interactive communication comes in various flavors, from social video chats over the Internet (e.g. Skype video, Apple Facetime) to immersive telecollaboration environments offered by many enterprises (e.g. Cisco Telepresence). Recent advances in communications and video compression technologies like H.264 [1–4] and its scalable extension called scalable video coding [5, 6] have made such multimedia applications possible through high compression efficiencies that offer a rich multimedia experience at much reduced bitrates.

Enabling compelling services, such as video conferencing and streaming, is a challenge due to the high demands that these systems place on the network. The quality of experience (QoE) offered by these services depends heavily on the characteristics

of the underlying network. There must be sufficient network resources available since the interactivity and performance of these applications is heavily degraded by network impairments such as packet loss, delay, jitter, non-availability of bandwidth, etc. Abundance in network resources is difficult to achieve in a best-effort network like that of the Internet. Moreover, client heterogeneity adds to the complexity of the streaming process. Client devices including mobile phones, PDAs, netbooks, laptops, workstations, IPTVs, etc. vary widely in their operating environment, computing power and display capabilities. They connect via heterogeneous access networks like residential broadband connections (DSL and cable), WiMAX, 3G, university campus and corporate networks. To deliver a high quality of experience to such a variety of clients (or participants in case of a video conferencing session), it is necessary for the video content to adapt its bitrate to the changes in bandwidth and client limitations. This will help in achieving a graceful degradation when network conditions deteriorate. Content adaptation must be done at a fine granularity to ensure the best video quality possible. It should be scalable to serve a large number of clients in real time and the reaction speed should be high enough to enable adaptations to quick bandwidth changes. The problem is more interesting when streaming 3D content, which requires twice the bandwidth since two bitstreams are transmitted (one for each eye) to each client and the added dimension of depth perception poses special challenges.

Our research work investigates this important problem of video content adaptation to varying network resources and client limitations using the scalable video communications approach. In scalable video coding (SVC), multimedia content is encoded in a set of layers providing temporal, spatial and quality scalability. The base layer provides a minimum acceptable level of video quality and each additional enhancement layer provides incremental quality improvements. SVC's layer structure allows the extraction and decoding of partial bitstreams at reduced resolutions. This property of SVC has led to its use in a number of applications including video

streaming [7,8], video conferencing [9], IPTV services [10,11], mobile TV [12,13], etc. Adaptation of scalable video to changes in network conditions in these applications forms the core of our research work and this thesis. Each of these applications differ in a number of ways in terms of network requirements and the end-user expectation of QoE. For e.g., video conferencing is tightly constrained by end-to-end delay and jitter constraints apart from real-time encoding. The users expectation from a video conferencing application is the ability to converse seamlessly. Streaming techniques on the other hand do not have jitter requirements, but the QoE expectations from user is very high in terms of spatial quality, frame rate, etc. When it comes to 3D streaming, the QoE depends heavily on perceived depth than on the quality of the individual views that make up the 3D video. Hence, we focus on each application individually and solve the problem of SVC-encoded content adaptation to varying channel conditions.

Solution in terms of extraction algorithms that maximize the reconstructed video quality for a given bitrate is proposed for each application. For the streaming scenario, a rate distortion optimal algorithm is developed for the extraction of MGS quality layers [5] from the SVC bitstream to adjust its bitrate to the current available bandwidth in the channel. Here, the extraction decisions are made over each GOP of compressed video data. For video conferencing, the RD optimal extraction decisions are made over a pair of frames to meet the tight end-to-end delay and jitter requirements. The proposed extraction algorithms for 3D content streaming maximize the overall perceived 3D quality based on human stereoscopic perception. When compared to current extraction methods, the new algorithms offer better video quality at a given bitrate while performing lesser number of metadata computations in the post-encoding phase. The solutions proposed for each application achieve the recurring goal of maintaining the best possible level of end-user quality of multimedia experience in spite of network impairments.

In the next sections, we summarize the research objectives, the key contributions of our work and the organization of this thesis.

1.1 Research Objectives

The key research objective of this work is to investigate scalable video communications and propose solutions that enable video content adaptation to varying channel conditions and client limitations in a variety of multimedia communication environments such as streaming, conferencing and 3DTV. The goal is to maintain the best possible level of end-user multimedia experience in spite of network impairments. The adaptations performed must maximize the reconstructed video quality and must be able to operate in real-time environments. The solutions must also satisfy application specific constraints as summarized below:

1. **Video Streaming:** For one-way streaming applications, rate-distortion optimal extraction of SVC bitstreams should maximize the video quality and minimize the delay incurred in metadata computations. It should aim at reducing the number of decodings performed while evaluating each layer's contribution to overall distortion minimization. This enables the extraction technique to operate in real-time, which is necessary for it to be used in streaming applications.

2. **Video Conferencing:** For two-way conferencing applications, the extraction of conversational video sequences should maximize the video quality while operating within the tight end-to-end delay and jitter constraints. The extraction should be rate-distortion optimal and should not incur any additional delay in the system.

3. **3DTV:** For streaming of 3D content in full-resolution stereo mode, the extraction technique should optimize the overall perceived 3D video quality rather than individually performing an RD optimal extraction on each of the left and

right-eye views.

1.2 Key Contributions

The key contributions of our research work and this thesis can be summarized as:

1. **Video Streaming:** For one-way streaming applications, a rate-distortion optimal extraction algorithm is proposed that maximizes the video quality. Extraction decisions are made over a window of one group of pictures (GOP) of compressed video data. It uses a combination of metadata computations and prediction mechanisms to evaluate the quality contributions of each of the layers in the bitstream. When compared with the current state-of-the-art techniques, the proposed algorithm achieves better video quality at a given bitrate while performing a lesser number of quality metadata computations.

2. **Video Conferencing:** For two-way conferencing applications, multiple extraction algorithms with different decision window sizes and jitter compensation requirements have been proposed. Due to the tight end-to-end delay and jitter constraints, the extraction decision window is limited to one or two frames in a GOP and is based on the importance of each layer in minimizing the distortion of the reconstructed video. The proposed technique of paired-frame extraction using quality metadata information performs an RD optimal extraction and provides better video quality than content-independent extraction techniques.

3. **3DTV:** For streaming of 3D content in full-resolution stereo mode, the proposed extraction technique takes advantage of the human brain's stereoscopic perception and optimizes the overall perceived 3D video quality by unequally allocating bits among the two views. Compared to equal bitrate allocation, the proposed technique achieves higher subjective quality.

1.3 Organization of the Thesis

The thesis is organized as follows: Chapter 2 describes the origin and a brief history of multimedia communication and video content adaptation mechanisms. It analyzes the effects of bandwidth on multimedia quality and explains the mechanism behind SVC along with brief descriptions of other bitrate adaptation mechanisms. Chapter 3 focuses on extraction algorithms for SVC-based streaming. Using three-screen TV as an application, the extraction problem in formulated. Solutions are proposed and validated through experiments and results. Chapter 4 proposes extraction algorithms for SVC-based video conferencing. The organization is similar to Chapter 3 in terms of application-level motivation, problem formulation, solution and results. Chapter 5 focuses on extraction algorithms for SVC-based 3DTV. It starts with the descriptions of various content formats and 3D display types and is followed by the proposal of a 3D streaming architecture and human stereoscopic perception based extraction algorithm. The technique is validated through subjective quality evaluations. Chapter 6 concludes the thesis with possible directions to future work.

CHAPTER II

BACKGROUND AND HISTORY OF THE PROBLEM

In this chapter, we study the background and history of the challenges in multimedia communication and content adaptation. We start with a description of various modes of multimedia communication including one-way streaming and two-way conferencing. Then we describe the multimedia system protocols used in such communications. This is followed by a study of the impact of bandwidth on multimedia quality along with techniques to measure such quality. Next, we discuss the various adaptive source coding techniques with focus on scalable video coding (SVC). We look at the various dimensions of scalability such as temporal, spatial and quality scalability and analyze how they are designed in SVC. Finally, we discuss alternate video content adaptation mechanisms.

2.1 Multimedia Communication – Types and Protocols

In today's Internet age, multimedia applications like web streaming, live broadcasting, IPTV, mobile video, video-on-demand (VOD), video conferencing, 3DTV, etc. are enjoying exponential growth. All these multimedia applications communicate the media information from the point of content generation or storage to the end-user. Such a form of communication can be classified as two-way or one-way based on the direction of flow of the media content. Also, depending on whether the content transmitted is encoded in real time or is pre-recoded, the communication can be classified as live or on-demand [14].

2.1.1 Two-way Communication

Two-way multimedia communication is characterized by media (audio and video) flow in both directions (full-duplex mode). At each user, media is captured, encoded and transmitted in real time. Both forward and reverse channels must exist for such a communication to occur. All interactive forms of communication like video conferencing, telepresence and telecolloboration fall under this category. When the number of participants are more than two, this form of communication is referred to as N-way video communication where each participant interacts with the remaining $N - 1$ participants. To maintain a seamless interaction, ITU-T recommendation G.114 [15–17] suggests the following guidelines:

- Packet loss should be no more than 1%.
- One-way latency should be no more than 150 ms.
- Jitter should be no more than 30 ms.

Real-time encoding enables the adaptation of source parameters and error-resilient tools (like forward error correction) to changes in channel conditions. Two-way data flow enables feedback-based source coding. For an N-way interaction, encoding decisions must be based on feedback from all users. Challenges in building interactive communication systems arise due to the tight delay constraints that limit the computational complexity of the encoding process. Encoding tools optimized for higher compression ratios cannot be used if such tools are computationally complex or incur large encoding delays (e.g., B-slices). Hence, the overall compression efficiency of such encoders [2] is low.

2.1.2 One-way Communication

One-way multimedia communication, commonly referred to as streaming, is characterized by media transport from a server to one or more clients. Feedback and other

control information could be sent from the clients back to the server on the reverse channel, when such a channel exists. Services like IPTV, VOD and live streaming fall under this category. Depending on the transport protocol and the type of server used, streaming can be classified as web streaming (also known as progressive download) and true streaming [18].

In web streaming, media placed on a web server is downloaded by a client using HTTP/TCP protocols (e.g., youtube). While downloading, the client starts playing the media after waiting a few seconds for initial buffering. The video download rate is the maximum that is allowed by the network and the server, and it is independent of the bitrate of the compressed video being downloaded. Hence, when the network state deteriorates, playout is interrupted unless sufficient data has already been buffered, which requires a longer startup delay. However, live streaming applications cannot have a startup delay of more than a few seconds since media data is captured, encoded and streamed in real time. Moreover, HTTP uses TCP [19] as the transport layer protocol, which achieves data reliability at the cost of additional delay due to retransmissions. To ensure sequenced delivery, TCP does not hand over the newer packets to the application, even if they arrive on time, until the lost packet is recovered. This is not suited for multimedia transport where timely delivery is key to performance. Nevertheless, such web streaming techniques are popular for on-demand streaming of pre-encoded content.

In true streaming, multimedia content is delivered from a media server to clients via real time protocol (RTP) over UDP [20]. The date transfer rate is matched with the bitrate of the compressed audio and video streams. The server responds to changes in network conditions and feedback from clients by adapting the bitrate of the streamed media. This ensures a smooth playout even during deteriorating network conditions. True streaming is suited for broadcasting live events as the startup delay is minimal. The transport layer protocol used is UDP, which provides

no data reliability. Packet sequencing, loss detection and retransmission must be handled by the application at higher layers. This is advantageous since timely arrival is more important than reliability for real-time multimedia, which can tolerate some packet losses [20].

Advanced features like random access, fast forwarding and rewinding are provided with the real time streaming protocol (RTSP) [21]. Since the media data is delivered to the client application directly, users cannot easily download the entire media file, thus reducing copyright violations. Streaming of live content to a large number of geographically distributed users can be achieved through IP-multicast or content delivery networks (CDNs). For a smooth playout of streamed content, ITU-T recommendation G.114 [15–17] suggests the following:

- Packet loss should be no more than 5%.
- Latency should be no more than 4 to 5 s (depending on video application's buffering capabilities).
- There are no significant jitter requirements.

The delay constraints of one-way live streaming are more relaxed than those of interactive communication. Hence, live streaming uses coding tools like B-slices, which improve compression efficiency at the cost of higher computational complexity and delay. The size of the client group determines the employment of feedback-based encoding tools. Error-resilient coding techniques usage is adapted to network state variations in a manner acceptable by most users [2].

With the new wave of 3D movies and affordable 3DTVs, streaming of such content is becoming common. From the server's aspect, this means streaming of two videos, one for each eye, to each 3D content subscriber. Such transmissions might have additional bandwidth requirements, depending on whether the 3D content is represented in full-resolution stereo or frame-compatible mode [22, 23].

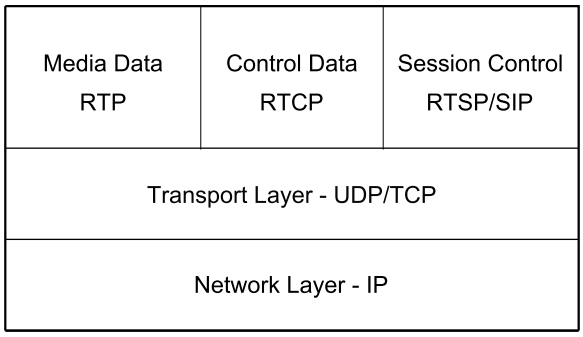


Figure 1: Protocols used in media streaming.

2.1.3 Multimedia Communication System Protocols

The protocols used in media streaming can be divided into three categories [24] as illustrated in Figure 1. They are:

1. *Network layer*: Protocols in this layer provide network-level functionalities like addressing, routing, etc. For Internet-based media streaming, IP [19] is used at this layer.

2. *Transport layer*: Protocols belonging to this layer provide end-to-end data transport services. UDP and TCP are the lower-layer transport protocols over which upper-layer transport protocols like RTP and RTCP [25] are implemented.

3. *Session control*: These protocols provide signaling messages for procedures like call setup, session initiation, etc. They control the delivery of multimedia data once the session has been established, and they also provide other interactive features like random access, fast forwarding and rewinding of streamed media. Examples include SIP (Session Initiation Protocol) [26], H.323 and RTSP [21].

RTP [20, 25, 27–29] provides an end-to-end transport function for real-time media delivery over an unreliable transport layer, such as UDP. RTP does not guarantee reliable delivery but provides mechanisms such as time stamps, sequence numbers,

payload type identifiers and source identifiers [24]. Media data is transported as the payload of the RTP packet. RTCP [25] is the associated control protocol that provides QoS feedback by periodic reporting of reception quality, participant identification and inter-media synchronization.

Other protocols used in media streaming include RSVP [30], which is used for reserving network resources to provide QoS guarantees to clients. Session announcement protocol (SAP) [31] and session description protocol (SDP) [32] are used to announce and describe ongoing sessions respectively.

2.2 Impact of Bandwidth on Multimedia Quality

The quality of user experience (QoE) with all media applications depends heavily on the characteristics of the underlying network that transports the media. Public networks such as the Internet are characterized by varying bandwidth conditions. There are no QoS provisioning mechanisms for real-time traffic on the Internet. The available bandwidth for a flow varies with time depending on the current link usage due to other flows.

In streaming and interactive video communication, playback begins while the media data is being received. The decoder waits for the receiver buffer to reach a certain level of fullness before decoding the first frame of video data. This waiting period is referred to as the startup delay at the decoder. For e.g., if the decoder buffers up N frames before starting the decoding process, then the startup delay is N/F seconds, where F is the frame rate. Once the decoding process has begun, the decoder continuously decodes frames from the buffer and displays them at a constant rate of F. Hence, if the network does not deliver data fast enough to the receiver, it is possible for the receiver buffer to underflow. In the above example, the maximum tolerable network congestion period for which the decoder can decode and display frames without receiving any additional data from the network is equal to the start

up delay of N/F seconds. Once the congestion exceeds this duration, the decoder runs out of data to decode and playback is interrupted until further data is received. Such interruptions can be reduced if the decoder waits for a longer time before starting the decoding process. The longer startup delay allows enough data to be buffered so that when the network is highly congested, the receiver buffer will not go empty and hence the decoding and display process can continue smoothly. However, to meet latency constraints, live streaming limits the startup delay to a few seconds [15–17]. This is especially true when users switch frequently between multiple streams where this delay will occur at every switch. This forces the decoder to start decoding even if the buffer has not reached the desired level. This problem is even more challenging for interactive video communication where there is no startup delay and decoding begins immediately after receiving the first couple of frames of video data. By reducing the time spent by media data in the receiver buffer, the end-to-end delay of a video frame is kept at a minimum level so that interaction can be maintained.

Under varying network conditions and such tight startup delay constraints, the problem lies in maintaining the best possible end-user QoE. This requires delivery of maximum video quality (in a rate-distortion sense) allowable by the network bandwidth while ensuring smooth playback without interruptions. A graceful quality degradation to deteriorating network conditions can be achieved if the server or one of its proxies adapt the video bitrate to match the current available bandwidth.

The above approach can be summarized by two main operations:

1. Observing the characteristics (e.g., available bandwidth) of the channel between the server and the client.

2. Maximizing the video quality delivered to the users by adapting the source bitrate of the multimedia stream as a reaction to the changes in network. This requires knowledge of the relative importance of various portions of the bitstream in terms of their contribution to the reconstructed video quality.

2.3 Video Quality Measurement

Digital video quality can be measured by a number of objective and subjective metrics. Objective metrics report the video quality based on a computational model that takes into consideration both the source and the processed video. These metrics can be broadly classified depending on the amount of information it uses about the original source video. Full reference metrics such as PSNR (Peak signal to noise ratio) need the complete original source video. It computes the mean square error between pixels on a frame-by-frame basis. PSNR measurements are easy to carry out, sensitive to small changes in pixel values and accurately reproducible. The actual value is not definitive, but the comparison between two values gives a commonly understood measure of quality. They are very useful when comparing different encoding or extraction algorithms [33]. Other type of metrics include reduced reference metrics which have limited dependencies on the source. No-reference metrics do not have any dependence on the source video. They evaluate the video quality from the processed video alone [34].

Subjective metrics include metrics such as mean opinion scores (MOS) and mean time between failures (MTBF) [35, 36]. MOS refers to a scoring scheme where the subject is asked to rate a video by choosing a number within a specified range. The lowest and highest endpoints in the range refer to the lowest and the highest quality video in the test database. MTBF is a functional quality metric where failure refers to video artifacts deemed to be perceptually noticeable. These two metrics have similar standard deviations across video stimuli and subjects. In our past work, we have used the real-time AVQ meter [37–39] to provide an objective estimate of MTBF for transcoding experiments [40] and video streaming experiments [41]. We have developed a taxonomy of visibility of artifacts and classified them as compression artifacts (CA) and network artifacts (NA) depending on whether they occur due to encoding or due to packet losses [42]. In our streaming and transcoding experiments,

we simulated conditions that generated such artifacts and used them to verify the performance of the AVQ metric.

2.4 Adaptive Source Coding Techniques

Client devices of a streaming or a conferencing system include mobile phones, PDAs, netbooks, laptops, workstations, IPTVs, etc. These devices vary widely in their operating environment, computing power and display capabilities. They connect via heterogeneous access networks like residential broadband connections (DSL and cable), WiMAX, 3G, university campus and corporate networks, which vary in available bandwidth. To deliver a satisfactory multimedia experience to such a variety of users, it is necessary to adopt video technologies that can adapt to each user's network connectivity and operating environment limitations. It should provide a mechanism for adjusting the transmitted bitrate of the video in response to changes in the network state like available bandwidth conditions. Scalable video coding (SVC) [5,6,43,44], is one of the key technologies that enable this adaptation. It was recently standardized as the scalable extension of H.264/AVC [45]. SVC is used in a number of applications including video streaming [7,8], video conferencing [9], IPTV services [10], etc.

2.4.1 Scalable Video Coding (SVC): Bitstream Extraction

SVC is a layered coding technique. The video content is encoded at different spatial, temporal and quality resolutions, which are arranged into a base layer and a set of enhancement layers within a single bitstream. The enhancement layers of a frame are predicted from its base layer, and hence it is essential to decode the base layer at all times. To increase the compression efficiency, SVC uses a number of inter-layer prediction mechanisms. To decode a particular layer, all layers up to that layer in that dimension (spatial/temporal/quality) must be decoded to satisfy the inter-layer dependencies. The bitrate of an SVC stream is far lesser than the sum of the bitrates of all the individual substreams put together. A complete SVC system [46] is shown

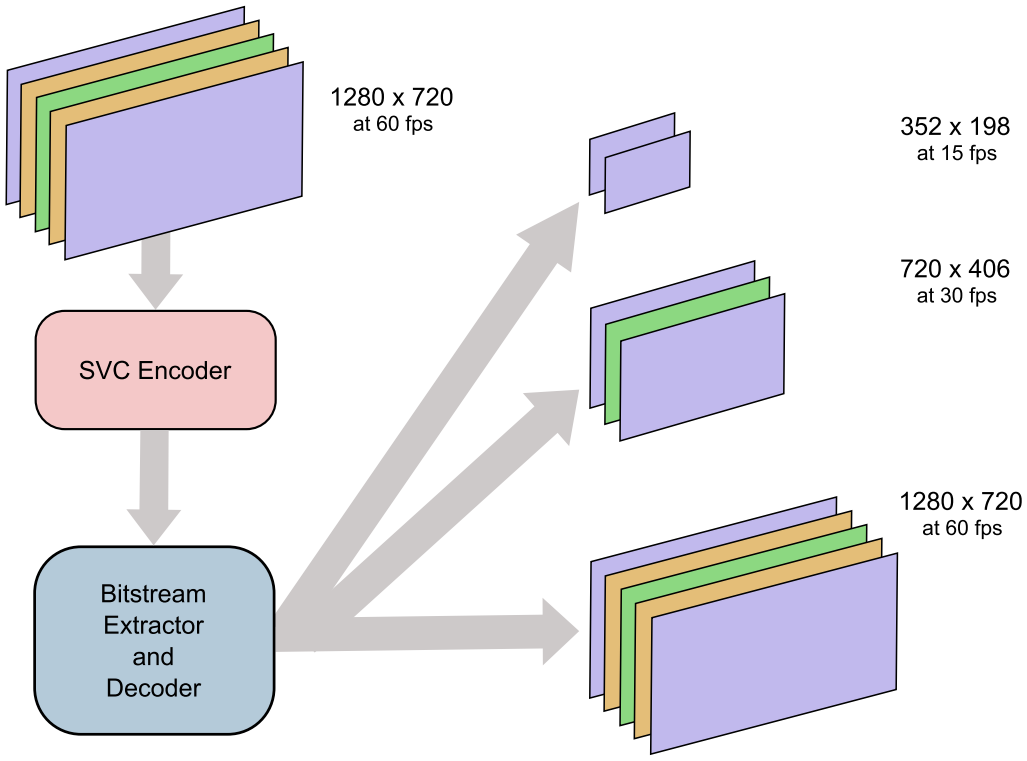


Figure 2: An SVC system: encoder, extractor and decoder.

in Figure 2.

The main advantage in using SVC for streaming and interactive video communication is its support for partial extraction and decoding of bitstreams at a lower temporal, spatial and quality resolution. On decoding, the base layer provides a minimum acceptable level of video quality and each additional enhancement layer provides incremental improvements. When the network condition deteriorates, only the base layer is extracted. Depending on the available bandwidth, additional enhancement layers are extracted and decoded, thus improving the video quality at the client. In this way, the stream's bitrate is adjusted dynamically and graceful degradation is achieved when bandwidth drops. The degree of scalability is at a frame level (MGS scalability, [5]), i.e., the number of quality layers extracted varies frame by frame. The design of temporal, spatial and quality scalability in SVC is described in the following subsections.

2.4.1.1 Temporal Scalability

SVC provides temporal scalability by partitioning the set of frames in a GOP into a temporal base layer and a set of temporal enhancement layers. This type of scalability is not new in SVC since it already existed in H.264/AVC [5] where such a scalability was achieved using hierarchical prediction structures as shown in Figure 3. The base temporal layer ($T = 0$) is encoded as a P-picture and the higher temporal layers are encoded as B-pictures. Prediction of a B-picture is allowed only from a lower temporal layer picture and hence, scalability is achieved by partial decoding of temporal layers starting from the zeroth layer. Other prediction structures are also possible, for e.g., zero-delay prediction structures using simply P-pictures at all temporal layers is used in video conferencing applications since the structural delay is zero.

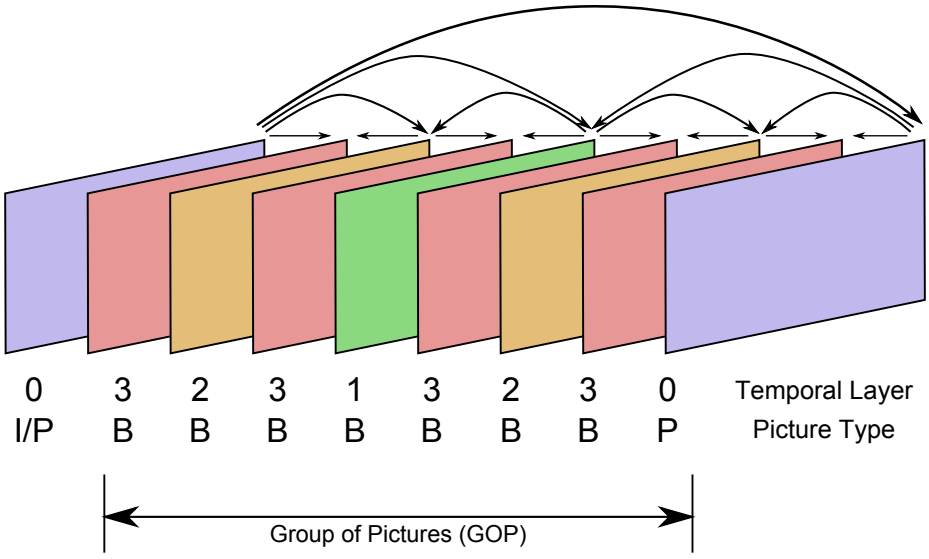


Figure 3: Temporal scalability in SVC using Hierarchical B-pictures.

2.4.1.2 Spatial Scalability

Encoding a bitstream into a number of spatial layers involves motion-compensated prediction and intra-prediction at each spatial layer. In order to improve the compression efficiency, SVC incorporates a number of inter-layer prediction mechanisms [5,47] as illustrated in Figure 4. Switching between spatial layers can occur only at IDR pictures since spatial scalability is designed to operate with a single motion compensation loop running at the target spatial layer (single-loop decoding).

1. **Inter-layer motion prediction:** Using a new macroblock type (signaled through the base mode flag), only a residual signal is sent for the higher spatial layers. The reference pictures, motion information, etc., are derived from the co-located macroblock of the lower spatial layer. For conventional macroblock types, SVC includes the option of using the lower spatial layer motion vectors as predictors for the motion vectors in the higher spatial layers.

2. **Inter-layer residue prediction:** This can be used for all inter-coded macroblocks. The residual signal from the lower spatial layer is blockwise upsampled

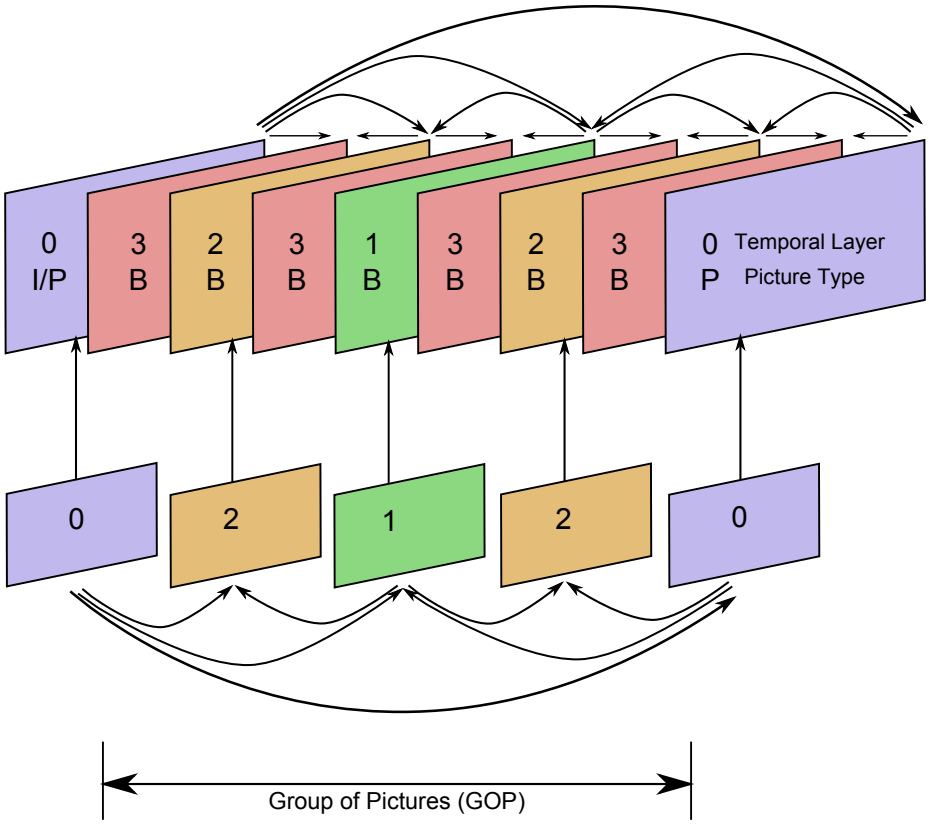


Figure 4: Spatial scalability in SVC using inter-layer prediction mechanisms among the two spatial layers.

and used as a predictor for the residue in the higher spatial layers.

3. **Inter-layer intra prediction:** When the base mode flag for an enhancement layer macroblock is set and the region in the base layer is coded using intra prediction, then the prediction for the higher spatial layer macroblock comes from inter-layer intra prediction. It is required that the intra blocks in the base layer are coded using the contrained intra-prediction mode so that it is not required to decode inter-coded macroblocks in the base layer, which would require a separate motion-compensation loop for the base layer.

2.4.1.3 Quality Scalability

Quality scalability implemented as spatial scalability with identical spatial resolutions at both the base and enhancement layers is known as coarse-grained quality scalability (CGS). The disadvantages with CGS approach is that the number of supported rate points are very limited and the switching can occur only at IDR pictures. SVC provides a better mechanism of quality scalability called medium-grain quality scalability (MGS) [5]. It provides a wide range of rate points and switching can occur at any frame. For every frame, there can be a number of MGS quality layers, each having a set of enhancement layer transform coefficients. This gives the ability to skip any enhancement quality layer from a frame. To improve the compression efficiency, SVC predicts the base quality layers from the highest quality reconstructions of their parents. However, this leads to a drift between the encoder and decoder since throwing away quality layers from a frame affects its reconstructed video quality and when used as a reference for future frames, it leads to a different motion compensation loop at the encoder and decoder. To compensate for this problem, SVC includes a new concept of encoding key pictures. Such pictures are always predicted from the lowest base quality layer reconstructions of their parents. Frames at temporal layer zero are usually encoded as key pictures and this ensures that drift is contained within a GOP and not propagated beyond the temporal layer zero picture.

The structure of SVC NAL unit header is shown in Figure 6. The header is four bytes in length. In the third byte of the header, the three bits labeled DID represent the syntax element *dependency_id* and indicate the spatial layer of the NAL unit. The next three bits (QID) represent the syntax element *quality_id* and indicate the quality layer to which the NAL unit belongs. The next three bits (TID) represent the syntax element *temporal_id* and indicate the temporal layer of the NAL unit. These three fields are the most essential fields in the SVC NAL unit header since they enable partial stream extraction along the spatial/CGS, quality/MGS and temporal

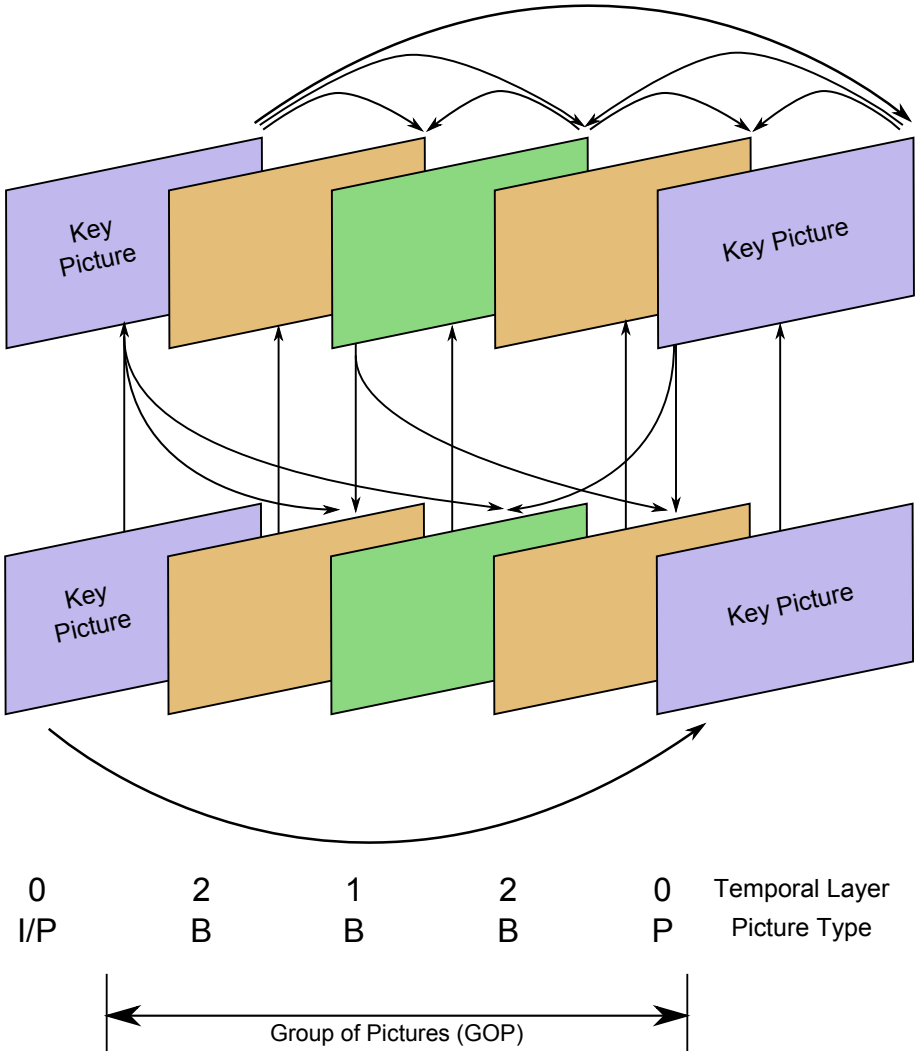


Figure 5: MGS Quality scalability in SVC using key pictures.

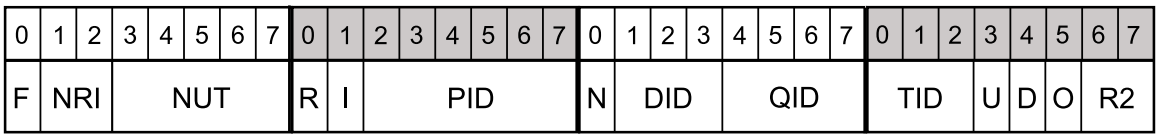


Figure 6: Four byte SVC NAL unit header.

dimensions without parsing the entire bitstream. The PID field is 6 bits long and represents the syntax element *priority_id*. It can be used by a post-encoding process to assign priority levels to the various spatial, quality and temporal layers in the SVC bitstream depending on the contribution of each layer towards reconstructed video quality. This information will be of great help to the extractor, which could be located at an intermediate network node, in extracting layers in a rate-distortion optimal way so that maximum QoE can be ensured under the current available bandwidth conditions. The remaining header fields indicate other information about the NAL unit [48] such as the payload type, inter-layer prediction mechanisms used, etc. The extraction delay is minimal in SVC since the layer information needed for extraction is located in the header [45, 48] of each NAL unit.

Other technologies suited for bitstream's rate adjustment include rate control at the encoder, multiple bitstream switching [49], transcoding [50, 51] and multiple description coding [52, 53]. The pros and cons of each of these techniques along with reasons for their unsuitability in real-time communications has been discussed in detail with respect to each application in the later chapters.

CHAPTER III

SVC BITSTREAM EXTRACTION FOR STREAMING

This chapter focuses on bitstream extraction techniques for scalable video coding (SVC) based multimedia streaming. The challenges involved in multimedia streaming are explored using the three-screen TV application as an example. An end-to-end system architecture based on an SVC home gateway for television broadcasting services (like cable/satellite/IPTV) is described. Next, the problem of adapting the bitrate of SVC encoded video to varying channel conditions is formulated with an aim of maximizing the reconstructed video quality. Current state-of-the-art solutions to this problem are discussed and their pros and cons are evaluated. This is followed by our solution design and algorithm, which is validated through a variety of experiments. Comparison with existing techniques show that our algorithm achieves better video quality for a given available bandwidth in the channel while minimizing the number of quality metadata computations performed in the post-encoding phase. This makes our extraction strategy highly suitable for real-time media streaming applications. Finally all our findings and results for the media streaming application are summarized.

3.1 Application – Three-Screen TV

Three-screen TV refers to the displaying of video content on multiple devices (usually two to three), each with different screen size and spatial resolution. In the broader sense, it also includes devices operating under different environments, network connectivities, processing power, etc. The challenge lies in adapting the video content to each of the device's characteristics while maximizing the video quality in each case. With the advent of smart phones and tablets that are capable of decoding and playing

video streams in real-time, the concept of three-screen TV has gained more practical importance, the three screens being those of the HDTV, tablet or a laptop, and a mobile phone. Each one of these devices operate under different power limitations and network connectivities: HDTVs usually have a wired connection, tablets operate with a WiFi or ethernet connection, mobile phones operate with a 3G, WiFi, or a bluetooth connection. Service providers aim to deliver content to each of these devices in a way that is suited to the device's operating environment so that the quality of experience (QoE) can be maximized on each of the three screens. The straight forward way of achieving this goal would be to send multiple bitstreams (the exact number depends on the number of devices) representing the same visual content to each subscriber. However, this proves very expensive for the service provider as it requires allocation of additional network resources such as bandwidth. Moreover, these streams cannot adapt to varying network conditions such as bandwidth fluctuations, changes in packet loss rates, etc. Hence, the service providers require a way to send a single video stream that can be scaled so that different devices can decode different portions of the stream depending on their operating conditions and available bandwidth. Scalable video coding forms an ideal solution to this requirement. It eliminates the need to encode multiple streams at different bit-rates and spatial resolutions. The rate-distortion performance achieved is comparable to that of non-scalable streams and it has the additional advantage of allowing extractions at reduced bitrates, thus adapting the video stream to changes in network conditions. This helps achieve a graceful degradation when the available bandwidth in the network drops. However, this convenience comes with a minor increase in the encoding time and complexity when compared to non-scalable streams.

The real power of scalable video coding is the ability to extract streams at reduced bitrates that offer a reduction along the spatial, temporal or quality dimensions. The nodes where the extraction process is performed is usually termed as an adaptation

point [54]. Such points are strategically positioned in the video distribution chain so that they can respond to changes in network conditions and operating environment promptly. If this adaptation point is placed at the client device, then it will be useful in only reducing decoding complexities and will not help achieve graceful degradation with changes in bandwidth since the complete scalable stream would have already been delivered to the client. Adaptation point at the headend (server) is not a good choice either since now multiple streams must be sent through the entire distribution chain, which negates the purpose of using scalable video streams. Hence, the best choice for adaptation and bitstream extraction is at an intermediate node, usually closer to the client and the last-mile access network where bandwidth fluctuations are maximum and packet losses occur more frequently. Having the adaptation point closer to the client ensures scalability of the system since only one stream travels through most of the network path. It also makes sure that graceful degradation is achieved by adapting the stream to changes in bandwidth in the last mile.

The key step in adapting a bitstream to changes in channel conditions is to evaluate the available network resources and decide what portions of the stream to extract. The node that does this process is called a decision agent. It is important to know that the decision agent and the adaptation point need not be the same node in the distribution chain. The decision agent can request the portions of the stream it needs and the adaptation point complies with this request. Let us examine the case where the client acts as the decision agent. In such cases, the available bandwidth is first measured by the client device. Based on average bitrates of the individual layers in the SVC stream, the client subscribes to the adaptation node for a set of layers. The number of layers subscribed is refreshed from time to time as the channel conditions change. The average layer bitrate information is sent to the client at the beginning of the streaming session. Such a subscription-based technique suffers from a number of shortcomings, such as:

1. Individual video frame data is known for its burstiness, and it might not adhere to the average layer bitrates.

2. Layer switching at the adaptation point is done at the same granularity at which the request is made, which depends on how frequently the client can measure its available bandwidth. This is typically of the order of a few seconds, could be more for portable devices with limited computing resources. Layer switching at this granularity reduces the effectiveness of SVC, which allows switching within every frame (of the order of few tens of milliseconds).

3. Since the adaptation is performed after receiving the decision from the client, there will be an added latency in implementing the bitrate adjustment of the stream. Hence, the adaptation speed is reduced. This affects system performance when available bandwidth changes at a rate faster than the adaptation speed.

Hence, clients acting as decision agents are not suitable. The next choice is to choose the headend (server) as a decision agent. But, the main disadvantage here is that since it is the farthest from the client, it may not have reliable information about the last-mile access network which is the key factor that degrades performance and QoE. Hence, the best choice is to have an intermediate node in the distribution chain as the decision agent. This node must be closer to the client so that it can make reliable measurements of the last-mile access network state. Also, it must be powerful enough to make frequent measurements so that adaptation can be done on a fine scale (e.g. per-frame basis). To maximize the adaptation speed, the same intermediate node is chosen as both the adaptation point and the decision agent.

In the context of television broadcasting and video on demand (VOD) services, this intermediate node is known as the home gateway. The service providers (cable

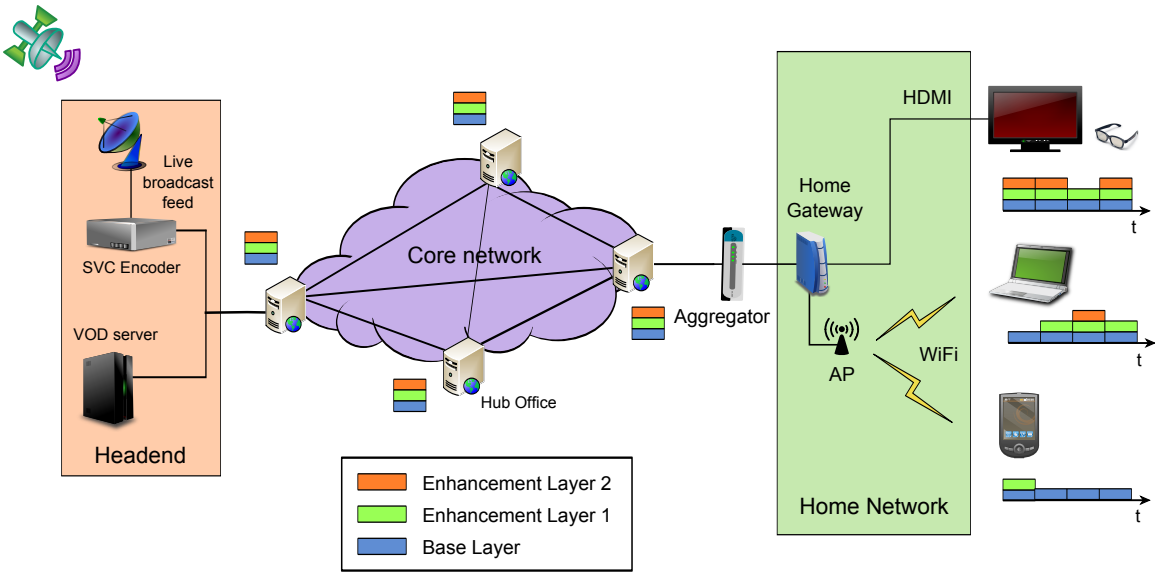


Figure 7: SVC home gateway based three-screen TV architecture.

TV/Satellite dish/IPTV) send a single scalable bitstream for each channel or on-demand content to the home gateway. The home gateway extracts the video bitstream depending on each of the currently active receiver's operating network conditions. For e.g., it could decode the complete bitstream for the HDTV, whereas it may decode only the base layer and a few enhancement layers for the tablet operating on a local WiFi. The gateway integrates the set-top box, modem and wireless access point functionalities so that it can deliver TV channels and VOD content to all the devices that have both wired and wireless connectivity. Such type of services are usually termed by the service providers as *TV anywhere within home*.

Figure 7 illustrates the end-to-end architecture of an SVC home gateway based three-screen TV architecture. The three main components of the distribution network are the headend, core network and the home network. This architecture is generic and is suitable for any type of distribution system including cable, satellite, IPTV, etc. The headend consists of equipment that capture live broadcast feeds and encodes them into SVC format in real time. VOD servers contain pre-encoded SVC streams that are fetched according to individual user's demand. All the content enters

the core distribution network of the service provider. It is distributed to individual homes through the access network via aggregators (e.g. DSLAMs in DSL networks and CMTS in cable networks). Within the home, the home gateway receives the scalable bitstreams. All the devices that need access to the SVC content (TV channels or VOD) register with the home gateway. As shown in the figure, the devices may include HDTVs, laptops, netbooks, mobile phones, PDAs with heterogeneous network connectivities (e.g. WiFi, bluetooth, etc.). The home gateway is made aware of the real-time decoding capabilities and other limitations associated with each device. Adaptation decisions are made at the home gateway after measuring the available bandwidth conditions between itself and each of the device at frequent intervals. This is followed by bitstream extraction and delivery to each device as an unicast stream. From the device's perspective, this involves installing a simple software that can receive, decode and display unicast IP streams.

3.1.1 Alternate Bitrate Adaptation Mechanisms

Other technologies suited for bitstream's rate adjustment include encoder-based rate control, multiple bitstream switching [49], transcoding [50] and multiple description coding [52, 53]. Since the encoding is done in real time, rate control of a single non-scalable stream can be done at the encoder by adjusting the encoding parameters like quantization step size, picture type, etc. Due to a large number of clients viewing a live stream, there is no one way to satisfy the bitrate requirements of all the clients simultaneously. A more practical option is encoding a fixed number of bitstreams at different bitrates and switching among them at the home gateway depending on the available bandwidth between the client and itself. Switching can occur only at designated points in the stream, such as I-pictures. However, I-pictures are used sparingly, as their frequent use reduces compression efficiency. This results in delayed switching that reduces the reaction speed to changes in bandwidth. The granularity

achieved is coarse depending on the number of streams the encoder can encode in real time (usually two or three maximum). Moreover, this technique involves multiple bitstreams at different bitrates traversing the network which is expensive for service providers, in terms of allocation of extra resources. On the other hand, SVC requires the encoding of a single stream that is slightly more complex than that of a non-scalable stream. Bitrate adaptation to changes in available bandwidth is handled separately at the home gateway.

Transcoding at the home gateway nodes is another common choice. It involves at least partial decoding of the stream and re-encoding the stream at a different bitrate depending on each of the client's requirements. Such operations are computationally expensive and incur high delays and hence, not suitable for live broadcast of TV channels [55]. In multiple description coding (MDC) approach, the content is encoded into multiple descriptors, each of which is independently decodable. When more than one descriptor is received, the quality is enhanced. MDC's success depends totally on path diversity [56,57], which rarely exists in the last-mile access network between the home gateway and the device. Also, the redundant information among the multiple descriptors reduce the compression efficiency. Hence, it is not suited for the three-screen TV application.

The performance of SVC based systems rely on the timely delivery of at least the base layer. In video streaming using SVC, additional error control techniques can be employed to ensure the safe delivery of base layer from packet losses. For e.g., the base layer can be protected with stronger forward error correction codes (FEC) [58] compared to the enhancement layers.

3.2 SVC Bitstream Extraction – Preliminaries

A scalable video stream consists of video content encoded into a number of temporal, spatial and quality resolutions within a single bitstream. The scalability is achieved

by providing the ability to pick out and decode partial portions of the stream corresponding to certain spatial, temporal and quality layers. This picking out process occurs in the compressed bitstream domain prior to decoding and is known as bitstream extraction. The main problem behind such an extraction process is how to extract a rate-distortion (RD) optimized bitstream for a given available bandwidth in the channel. In other words, the extracted video stream must be the best possible quality video stream (in a rate-distortion sense measured through a metric such as PSNR) that one can obtain at that bitrate. We attempt to solve this problem using SVC, which has been standardized by the ITU-T and the MPEG standardization bodies. This section studies the structure of an SVC bitstream, steps involved in an extraction process and then formulates the key problem of obtaining an RD optimal bitstream for a given bitrate that maximizes the decoded video quality.

3.2.1 SVC Bitstream Structure

Figure 8 shows the structure of an SVC bitstream with a GOP size of 8 frames. In the figure, each frame of video has been encoded into a number of spatial layers and within each spatial layer, it has been encoded into a number of quality layers. Each rectangular box represents a coded layer belonging to a specific spatial, temporal and quality layer. D (dependency ID) identifies the spatial layer, Q (quality ID) identifies the quality layer and T (temporal ID) identifies the temporal layer [48] of each coded layer. The frame numbers in the figure are in display order, i.e., the order in which the decoder plays back the video stream. All the spatial and quality representations are coded for one frame before encoding the next frame. The $Q = D = 0$ layer of each frame is termed as the base layer. The remaining layers are collectively called as the enhancement layers.

A scalable bit stream may contain a number of encoded spatial resolutions to serve devices with varied screen dimensions and decoding capabilities. For e.g., a

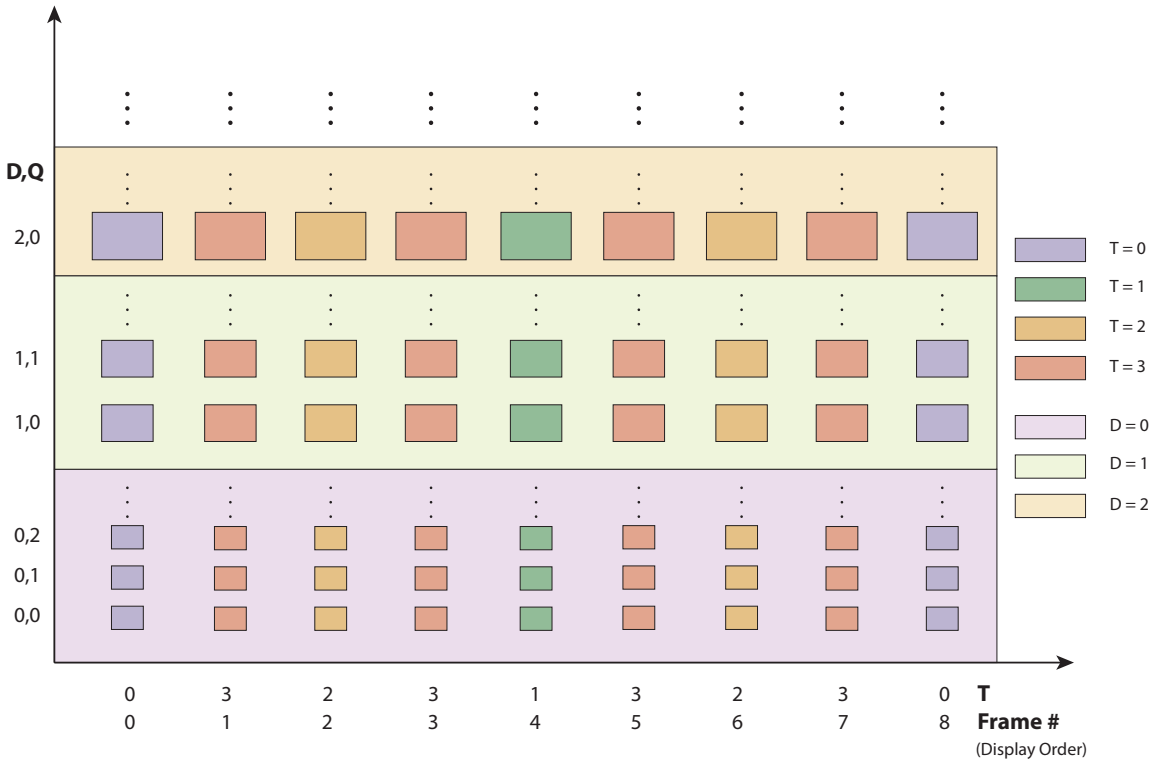


Figure 8: Structure of an SVC bitstream with a GOP size of 8 frames.

single SVC bitstream could consist of a 1920×1080 resolution stream for HDTVs, 1280×720 resolution stream for netbooks and a 720×480 resolution stream for mobile phones. These spatial layers are encoded in an RD optimal manner by utilizing the redundancies between them via inter-layer motion, inter-layer residue and inter-layer intra prediction mechanisms [5]. These spatial layers are numbered from zero for the lowest spatial dimensions and are incremented by one for every set of higher spatial dimensions. This value is represented by the *dependency_id* syntax element (D) [45] in the NAL unit header of each coded layer in the bitstream. This field is three bits wide. Hence, the maximum number of spatial layers that can be present in a bit stream is eight, with D varying from $0, 1, 2, \ldots 7$.

Within each spatial layer, a video frame is encoded into many quality layers, each of which is identified by the value of the syntax element *quality_id* (Q) in the coded layer's NAL header [45] . The layer with $Q = 0$ serves as the base quality layer for the

target spatial layer. The remaining higher quality layers (i.e. layers with $Q > 0$) are encoded as medium-grain quality scalability (MGS) layers using key pictures [5]. The higher quality layers are usually encoded with a finer quantization step size when compared to the base quality layer. SVC standard gives the ability to distribute these quantized coefficients into many quality layers so that decoding of each of these quality layers will give incremental improvements in quality. The *quality_id* field is four bits long in the NAL unit header. This limits the total number of quality layers to 16. Of these, the base quality layer is assigned a quality ID of zero. This leaves 15 quality layers for the enhancement layer at the target spatial resolution. Since SVC uses a 4×4 transform, there are 16 coefficients available in the enhancement layer. Hence, these 16 coefficients must be distributed among the available 15 layers. Even when 8×8 transform is used in high profiles, the standard still mandates that the number of quality layers in the enhancement layer be limited to 15. This is achieved by having a minimum of four transform coefficients in each enhancement quality layer.

The temporal layer information of the coded layer is represented by *temporal_id* field (T) in the NAL unit header [45]. The temporal layer of a frame represents its prediction capabilities and is dependent on the group of pictures (GOP) size. A group of pictures is the set of frames in between two temporal base layer ($T = 0$) frames along with the succeeding temporal base layer frame. In H.264/AVC, the concept of hierarchical B-pictures [59] was introduced, which provides temporal scalability. This is true for SVC, as well. Figure 9 shows the GOP structure using hierarchical B-pictures for a GOP size of 8 frames. The location of a frame in the hierarchy is represented by its temporal ID (T). Frames in the lowest level of the hierarchy ($T = 0$) are the most important since they predict the rest of the frames within that GOP. Hence, these are coded as I or P-pictures, thus forming the temporal base layer. The remaining members of the hierarchical structure are encoded as B-pictures with dependencies as shown in the figure. These B-pictures ($T > 0$) form the temporal

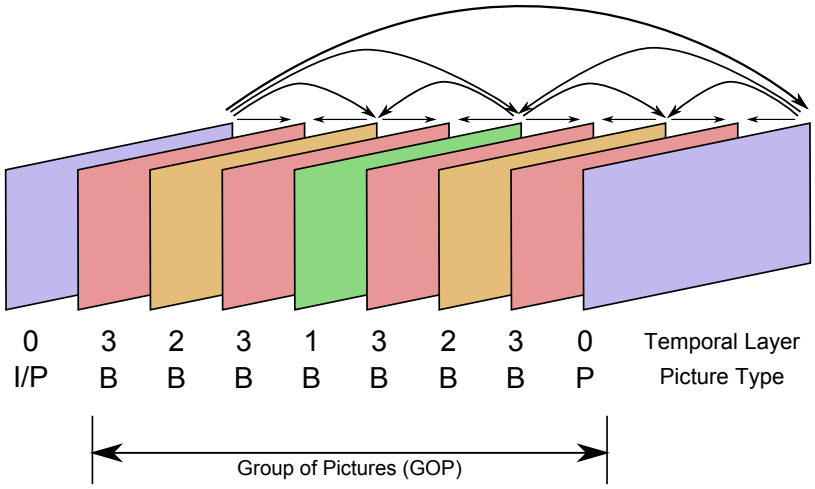


Figure 9: GOP structure using hierarchical B-pictures for a GOP size of eight frames.

enhancement layers. The example shown here is that of a dyadic regular prediction structure. Even nondyadic hierarchical prediction structures [5] exhibit the property of temporal scalability by the way the prediction structures are arranged. For a dyadic GOP structure with a size of N frames, the range of values of the temporal ID (T) can be described as:

$$T = 0, 1, 2, \ldots \log_2 N \tag{1}$$

To decode a layer with temporal ID $T = t$, it is necessary to decode all temporal layers with temporal IDs $T \leq t$. For dyadic prediction structures, this is evident from the prediction structure of Figure 9. To decode a stream at a quality layer $Q = q$ and a spatial layer $D = d$, it is necessary to decode all the quality layers $Q \leq q$ at the target spatial layer $D = d$. Also, the base quality layers from lower spatial resolutions ($D < d$) have to be extracted to satisfy inter-layer dependencies. SVC standard assures single-loop decoding i.e., any conforming bitstream is decodable with a single motion compensation loop. Hence, inter-layer intra prediction mechanisms are constrained to use only those macroblocks as reference that have been coded using constrained intra prediction. This eliminates the need for decoding any inter-coded macroblocks in the reference layer and hence, the stream can be decoded by running

a single motion compensation loop at the target spatial layer.

3.2.2 Bitstream Extraction: Problem Formulation

In a video streaming application, for e.g., the three-screen TV that we saw in the last section, the available bandwidth in the channel between the home gateway and each device varies due to a number of factors. The gateway responds by adapting the bitrate of the SVC stream to suit the current bandwidth conditions. This adaptation process involves extracting certain temporal, spatial and quality layers from the bitstream. Video streaming applications are, in general, more delay tolerant when compared to real-time conferencing applications [15, 17]. As long as the end-to-end delay is constant, the user would not experience any noticeable difference in QoE. As a result, adaptation decisions are performed over a GOP of frames, i.e., the extractor (home gateway in our example) waits to receive the entire GOP of coded data and then extracts the required layers from the GOP-sized stream. The choice of performing adaptation along a certain dimension (spatial, temporal or quality) within a GOP is governed by a number of factors as explained below.

The number of MGS layers selected can vary on a frame-by-frame basis, which allows a fine-tuned adaptation to changing bandwidth conditions. Of course, this introduces drift between encoder and decoder but the drift is contained within each GOP due to the encoding of key pictures in SVC [5]. Frame dropping (reduction of temporal frame rate) is the next means of adaptation at the extractor to changes in network state. This can also be done on a per-frame basis but has severe consequences than dropping quality layers. The decoder compensates for the missing frames by displaying the previous frame in its place. This results in stuck frames at the decoder output which reduces the QoE to a great extent, particularly with video sequences involving quick motion like sports, etc. Frame skipping is performed at the extractor by leaving out the base quality layers ($Q = 0$) of higher temporal layer pictures. This

happens when the bandwidth has fallen very low and already all the MGS layers at every temporal layer have been left out from extraction.

Switching spatial layers is the last resort at the extractor. When bandwidth falls to very low levels and the video's bitrate cannot be sustained at the higher spatial layer even at a reduced frame rate, the extractor switches to a lower spatial layer. Switching among spatial layers can only occur at IDR pictures since SVC decoders are required to run only a single motion compensation loop. This can cause switching delays that depends on the IDR period in the bitstream. IDR pictures are extremely inefficient from a compression point of view and hence, are encoded once every few seconds usually. This reduces the speed of adaptation to changes in bandwidth. Moreover, frequent spatial layer switches are a source of annoyance to the user if the multimedia viewing application does not have an automatic interpolation routine.

Initially, the target spatial and temporal layer is chosen by the client depending on its processing and display capabilities. This is communicated to the extractor prior to decoding. For e.g., in our three-screen TV application discussed previously, the devices register with the home gateway regarding their real-time decoding capabilities. Once these dimensions have been chosen, the base quality layers at the target spatial and temporal layer are extracted. The key decision to be made at the extractor is how to choose among the various MGS quality layers such that the resultant bitrate of the extracted stream does not exceed B bits/s, where B is the available bandwidth in the channel. The resulting video quality obtained by decoding the extracted partial bitstream needs to be the best (in an RD optimal sense) possible at the bitrate of B bits/s. In other words, we are trying to optimize the decoded video quality when a video is being streamed over a resource constrained channel, the resource in this case being the available bandwidth. The next goal in the extraction process is that the extracted substream should be a standards-conforming SVC stream, i.e., it should satisfy all layer dependencies and should be able to be decoded by any SVC-compliant

decoder.

When the bandwidth falls very low and the extractor cannot deliver the requested frame rate to the device, it begins to skip frames. At this point, all the MGS layers of all temporal layers have been eliminated from extraction. The problem takes a different form in such circumstances. Given the base quality layers ($Q = 0$) of all temporal layers in the GOP, how to extract the most RD optimal substream with a bitrate of B. Since there is hierarchy among the various temporal layers, the obvious choice of the order of extraction follows the increasing order of temporal IDs (T). But within a GOP, there are multiple frames belonging to the same temporal layer. Hence, the revised problem is to devise an order of extraction among frames at the same temporal layer.

Let us formulate the problem mathematically for a single spatial resolution. Let N be the GOP size in frames, Q_m be the maximum encoded quality layer, F be the frame rate and B represent the currently available bandwidth in the channel. Let G represent the number of GOPs per second. G can be computed as:

$$G = F/N \tag{2}$$

The bit budget (R_g) for the current GOP is computed as:

$$R_g = B/G = BN/F \tag{3}$$

The number of quality layers in a frame is $Q_m + 1$ since quality layers begin from zero. Hence, the total number of quality and temporal layers in the GOP (assuming there is one spatial layer) is calculated as:

$$L_m = (Q_m + 1)N \tag{4}$$

If each of the layers present in the GOP were given an absolute layer ID (L), then the range for L is $0, 1, 2, \ldots L_m - 1$. The value of temporal ID (T) is in the range $0, 1, 2, \ldots \log_2 N$. The value of Q is in the range $0, 1, 2, \ldots Q_m$. Let Size() represent

the function that computes the size in bits of its input argument. Let S be a layer in the GOP. It is represented by its temporal ID t, quality ID q and absolute layer ID l, i.e.,

$$S \equiv \{t, q, l\} \tag{5}$$

Let Δ represent the set of all layers S that when assembled together form a partial bitstream that represents the GOP of data and conforms to the SVC standard. Let $\Gamma(R_g)$ denote the set of all possible Δ that are of size less than or equal to the allocated bit budget (R_g) for the current GOP, i.e.,

$$\Gamma(R_g) \equiv \left\{ \Delta | \sum_{S \in \Delta} \text{Size}(S) \leq R_g \right\} \tag{6}$$

We are interested in the member Δ_{opt} belonging to $\Gamma(R_g)$ that minimizes the distortion function Dist(). This function computes the distortion (e.g. MSE) of the decoded stream represented by its input argument with respect to the source stream. It uses the available maximum quality reconstruction as the source stream while computing the distortions.

$$\Delta_{opt} = \underset{\Delta \in \Gamma(R_g)}{\text{argmin}} \ \text{Dist}(\Delta) \tag{7}$$

Equation (7) represents the problem of RD optimal bitstream extraction that we attempt to solve in the following sections. For the low bandwidth case, when no MGS layers are extracted, the set Δ is limited to layers L with quality ID $Q = 0$, i.e., the base quality layer of all frames in the GOP.

3.3 SVC Bitstream Extraction – Solutions

This section proposes the solution to the problem of RD-optimal bitstream extraction for a given available bandwidth in the channel. First we review the current state-of-the-art in SVC bitstream extraction. Specifically, we look at two recent techniques named JSVM–Basic and JSVM–QL and examine their pros and cons. Then, we describe the design considerations and motivation for our algorithm. We discuss how

the shortcomings of other techniques are overcome in our algorithm and what makes it suitable for real-time video streaming. This is followed by our algorithm overview and details with flow charts.

3.3.1 Related Work

Scalable video bitstream extraction has been studied by researchers since the standardization of SVC as the scalable extension of H.264/AVC [60–64]. Extraction trajectories based on subjective evaluation has been studied by the authors in [65–69]. Most of these perceptual quality based techniques result in coarse adaptation schemes with limited switching points as they do not leverage the MGS capability of SVC. The first bitstream extraction algorithm appeared with the reference software provided by the standardization body. This software is called the JSVM software [70] and hence, this technique came to be known as the JSVM–Basic technique. This technique is fairly straight forward. Initially, the base quality layers ($Q = 0$) of all the temporal layers at the lowest spatial layer ($D = 0$) are extracted. This is followed by the base quality layer of all temporal layers at the next spatial layer. This process continues till the base quality layer at all spatial layers have been extracted. Next, the higher quality layers are extracted. First, all the higher quality layers at the lowest temporal layer ($T = 0$) and lowest spatial layer ($D = 0$) are extracted. This is followed by the extraction of the higher quality layers at the next temporal layer at the lowest spatial layer. Once all the higher quality layers at each temporal layer at the lowest spatial layer have been extracted, the extraction moves towards the higher quality layers at the next spatial layer and the above process is repeated.

Figure 10 shows a simplified example of the above technique with a single spatial layer, five quality layers including the base quality layer and four temporal layers. The section shown here is the extraction of one GOP where GOP size is 8 frames. The numbers marked within the boxes are layer IDs (L) and represent the order of

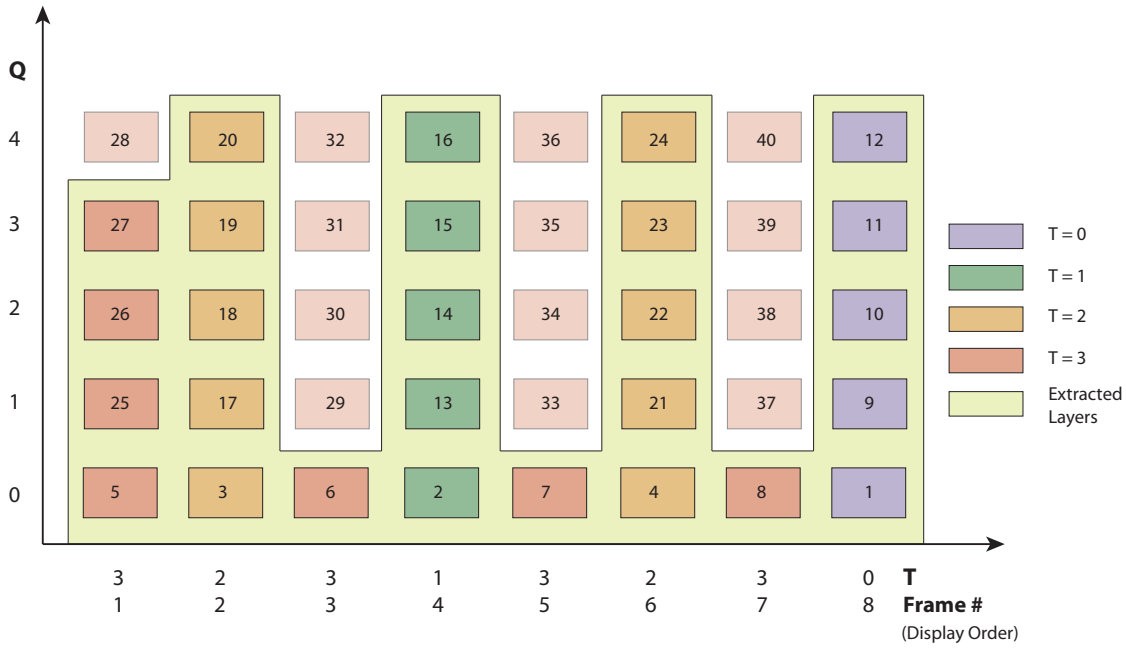


Figure 10: Typical order of layer extraction using the JSVM–Basic extractor for a GOP size of 8 ($T = 0, 1, 2, 3;\ Q = 0, 1, 2, 3, 4;\ D = 0$).

extraction of the layers. The numbering starts from the most important layers of the GOP, which is the base quality layer of each frame in the GOP. Among them, the ordering follows the temporal layer importance. Among frames with equal temporal importance (denoted by having the same temporal layer ID), the extractor chooses the layer that was received first in the GOP. Extraction starts at the layer marked 1 and proceeds in the increasing order of these layer IDs. After extracting each layer, the extractor checks whether the bitrate limit has been reached. If not, it continues the extraction process in the order of increasing layer IDs. When the limit is exceeded, it stops extraction and transmits the extracted layers. The shaded portion in the figure shows a typical extraction routine for a given bitrate. It can be noticed that before the higher quality layers of frames at $T = 3$ are extracted, all the higher quality layers of lower temporal layers have already been extracted. In this example, the bitrate limit was reached while extracting layer with layer ID 27, which is the quality layer with $Q = 3$ of the first B-picture at temporal layer 3. Hence, the higher quality layers

of none of the remaining B-pictures at $T = 3$ are extracted.

The pros of this technique is that it is simple to implement and can be done at an intermediate network node at equal ease for both live encoded content and on-demand content. The extraction delay is minimal since the decisions are made simply based on T, Q and D. This information is located in the NAL unit header of every layer. Moreover, no additional metadata computations are needed.

The main drawback in this technique is that the layers extracted are simply based on their quality, spatial and temporal IDs and not based on their actual contribution to reconstructed video quality. The extraction order is independent of the video content. For e.g., in Figure 10, the end quality contribution of MGS layers 1 and 2 of Frame # 5 could have been much more than the reduction in distortion achieved by sending the quality layers 3 and 4 of Frame # 2. Hence, from an RD optimality viewpoint, it would have been better to send the MGS layers 1 and 2 of Frame # 5 rather than the MGS layers 3 and 4 of Frame # 2. However, since this technique's choosing criteria does not minimize the distortion, it is incapable of such RD-optimal extractions. The layers extracted simply satisfy the bitrate constraint and produce a standards-conforming bitstream.

The drawbacks of the JSVM–Basic technique are overcome by the JSVM–QL technique proposed by the authors in [60]. This technique bases its bitstream extraction process on the quality-layers based framework. It is similar in concept to the quality layers used in JPEG 2000. The main aim is to identify the layers in the GOP that contribute towards maximum reduction of distortion. The layers are then extracted in the decreasing order of importance towards contribution to the reconstructed video quality. The overall technique consists of three parts: quality layers computation, quality layers signaling and quality layers based extraction. Quality layers computation occurs as a post-encoding process. It requires the SVC bitstream and the original source as its input. It decodes all the combinations of the temporal,

quality and spatial resolutions and evaluates the RD performance of each of those streams. Then the quality layer information is computed for each layer in the GOP. This is a very computation intensive process and not suited for real-time applications. The computed quality layers information is signalled through the bitstream either by using the *priority_id* field in the NAL unit header or by using the quality layer information SEI message. At the extractor, this quality layer information is read from the NAL unit header or the SEI message. This helps the extractor determine the RD importance of each layer, i.e., how much does each layer contribute to the reconstructed video quality. Based on this information, the extractor makes an informed decision and extracts layers in decreasing order of their quality contributions.

Though this algorithm proved better than JSVM–Basic in terms of extracting bitstreams with higher RD performance, it had many disadvantages as listed below:

1. The extraction is not perfectly RD optimal. The reason is that the JSVM–QL algorithm computes the impact of the refinement quality layers assuming that all the lower level quality planes are included. This results in inaccurate calculations of distortion reduction which reduces the RD optimality of the technique.

2. The algorithm performs an optimized extraction of only the MGS quality layers. For the base quality layers, its performance is identical to that of JSVM–Basic, which is characterized by content-independent and non-optimal extraction.

3. The extraction requires the quality layer information, which is computed as a post-encoding process and stored in the bitstream. This process is extremely computation intensive and not suited for real-time post encoding. For e.g., in the three-screen TV application, real-time SVC encoders at the headend capture the broadcast TV content from the satellite and encode them into SVC format. In such situations, the JSVM–QL technique cannot be used to insert quality

layer information into the stream since it is not possible to compute them in real-time.

3.3.2 Bitstream Extraction Algorithm

Rate-distortion optimal extraction involves the extraction of those layers that minimize the distortion in the reconstructed video. This requires the evaluation of the contribution from each layer towards reconstructed video quality. Once this information is available, bitstream extraction can be performed in the decreasing order of the layers' contribution to video quality, i.e., layers that reduce the distortion to the greatest level are assigned the highest priority and are extracted first and the process continues to lesser important layers and stops when either all layers have been extracted or the available bid budget has been reached. The bitstream produced as a result of this extraction process should still satisfy the layer dependency constraints and must conform to the standards.

The requirements of a good bitstream extraction algorithm can be summarized as follows:

1. Must perform an RD optimal extraction of bitstream for both base quality layer and MGS quality layers.

2. Must extract a bitstream that conforms to the SVC standard. It should satisfy all the layer dependencies.

3. Must minimize the number of decodings that need to be performed in order to evaluate the quality contributions of each layer in the post-encoding phase. This enables the algorithm to be used in real-time applications.

Figure 11 shows the block diagram of an end-to-end SVC system. The key blocks in the system include the process of encoding, post-encoding, extraction and decoding of bitstreams. An SVC encoder takes an uncompressed YUV stream as its input and

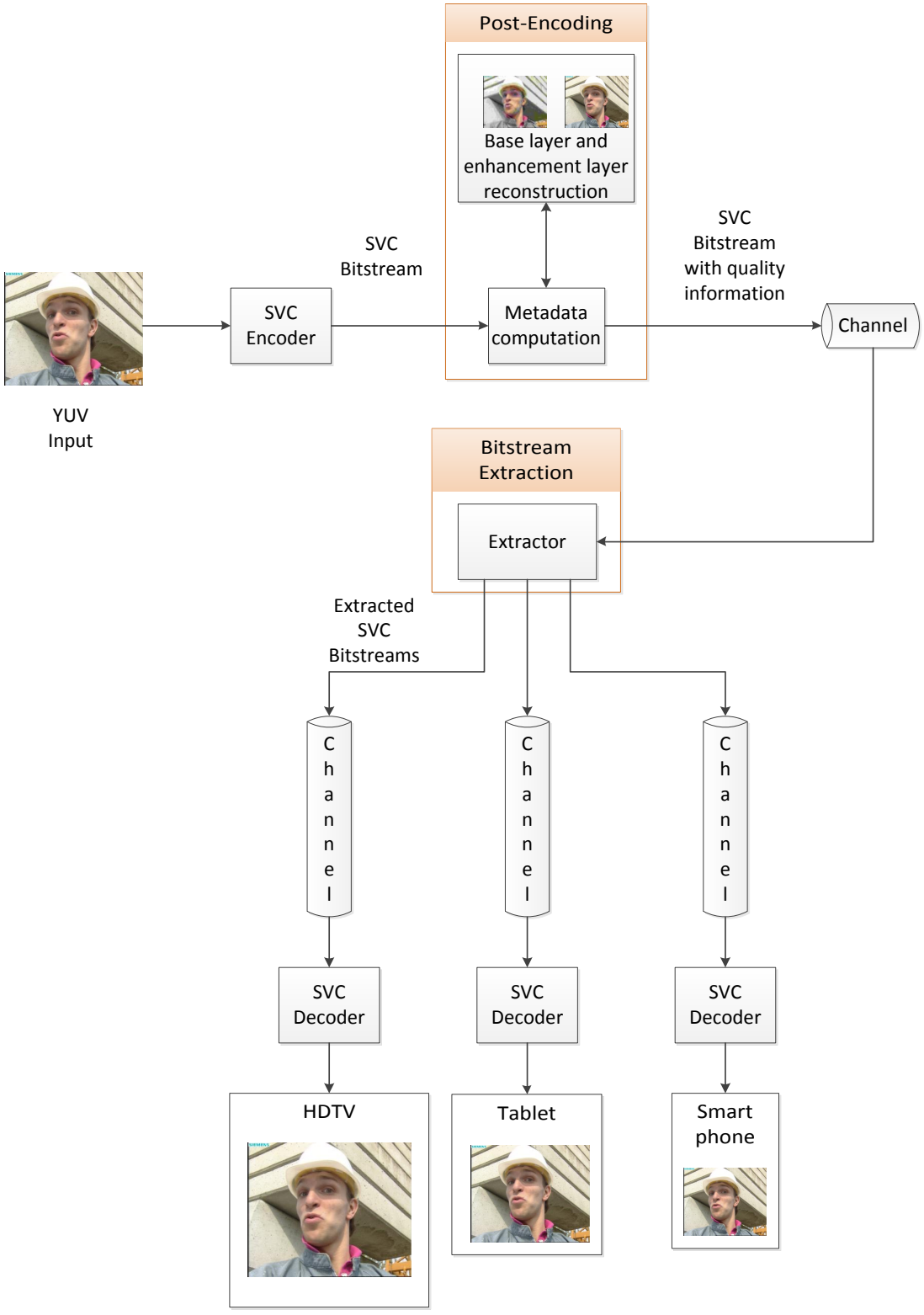


Figure 11: SVC-based streaming system: End-to-end block diagram.

generates an SVC bitstream with a certain number of quality, spatial and temporal layers, which has been set while configuring the encoder. This is followed by a post-encoding process where the stream is extracted at various layers and decoded in order to evaluate the quality contribution of each layer in the stream. The evaluation of the contribution of a specific layer towards reconstructed quality requires decoding the video stream with and without that layer and comparing the resulting video frames in each case with a reference frame and computing the distortion measure, for e.g. mean square error (MSE), before and after decoding that layer. The reduction in distortion obtained on decoding a layer is indicative of that layer's importance. Since decoding of bitstreams is involved in evaluating each layer's importance, carrying out this process at intermediate nodes along the network path (like the home gateway) will result in enormous end-to-end delays and hence, make the extraction process unsuitable for streaming applications. Hence, the layer quality contributions must be computed as a stand-alone process. For real-time encoded content, it is handled as a post-encoding process at the computationally powerful headend. Once computed, this quality information is stored in the NAL unit header of the bitstream or as SEI messages. This relieves the extractor located at an intermediate network node from decoding the bitstreams and computing quality contributions. By simply looking at the NAL unit header and the SEI messages, the extractor can identify each layer's importance and extract according to the available bandwidth in the channel. Once extracted, the video streams are transmitted on the network and are finally decoded by the end-user devices.

The blocks in Figure 11 that are improved by our algorithms are those of post-encoding and extraction (indicated in the figure by a container surrounding the block). The extraction algorithm consists of three components:

1. Computation of the quality contribution information of each layer in the bitstream. This is carried out as a post-encoding operation.

2. Signaling of this information in the bitstream.

3. Extraction based on this quality metadata at an intermediate network node.

Now, we describe the algorithm design for each of these components in detail. First we look at the computation of the quality information and priority ID assignment. Then, we describe how the metadata information is signalled in the stream and finally we propose the extraction algorithm that uses this quality metadata information.

3.3.2.1 Computation of Quality Metadata Information

The computation of layer quality contribution as a post-encoding process is usually computation intensive due to multiple decodings of the bitstream at various layer combinations. Given that SVC uses prediction pictures (P and B) to compress efficiently, the distortions computed fall into two category, namely independent and dependent distortions. Independent distortions are the ones where the reduction in distortion comes only from the current frame to which the layer belongs. This is true for pictures that have no children. For example, in Figure 9, the B-frames at the highest temporal level of 3 have no children and hence, decoding quality layers for this frame can contribute to the reduction of distortion only for this frame. In other words, decoding extra MGS layers for these frames do not have any effect on other frames. However, the same is not true for frames belonging to lower temporal layers such as $T = 0, 1, 2$. Additional decoding of MGS layers for these frames reduce the distortion not only for those pictures but also for their children since they are the prediction parents of their children. On additional decoding of an MGS layer for the parent, the prediction quality for the child is improved and hence, the quality of the child is improved as well. Exhaustively computing the reduction in distortion obtained by the additional decoding of each layer would involve methodic decoding starting from the first layer ($T = Q = D = 0$) and adding one layer at a time till all the layers have been added. This is a very time consuming and computationally

expensive process even for the headend or the server and cannot be performed in real time. The JSVM–QL technique described in the previous section, computes all these distortions and hence, is the reason it is not suited for real-time applications.

The solution to this problem is to reduce the number of decodings that need to be performed in order to compute the quality contribution for each layer. Our methodology is to perform a limited number of decodings at the lowest and highest quality layers for each frame within a GOP and predict the reduction in distortion that would be obtained on decoding the in between quality layers. Since the distortion at the two end points (the base quality layer and the maximum quality layer) is known, it is possible to estimate the distortions for the in between MGS layers. This involves distributing the reduction in distortion obtained by decoding the lowest and highest quality layers among the in between layers. Since all MGS layers do not contribute equally to reconstructed video quality, it is necessary to proportionately distribute the distortion reduction among the layers. In SVC [5], the quantized coefficients of the 4×4 or 8×8 transform are distributed into the various MGS layers at every spatial resolution. The order of distribution is in the block scan order and lower coefficients go into the lower MGS layers. This information is represented in the slice header. So if higher frequency components are present, there will be more non-zero values in the higher layers, increasing their size. If the higher MGS layers have a very small size, this implies that there are very few higher frequency coefficients, i.e., sharp gradients and edges are minimal in the frame and hence, good reconstruction quality can be obtained by simply decoding the lower MGS layers. In such cases, additional decoding of higher MGS layers do not contribute much to the reconstructed video quality. This implies that the size of the MGS layer at a particular spatial level is representative of its quality contribution. Hence, we distribute the reduction of distortion, from the lowest to highest quality layers, among the in between MGS layers in proportion to their sizes.

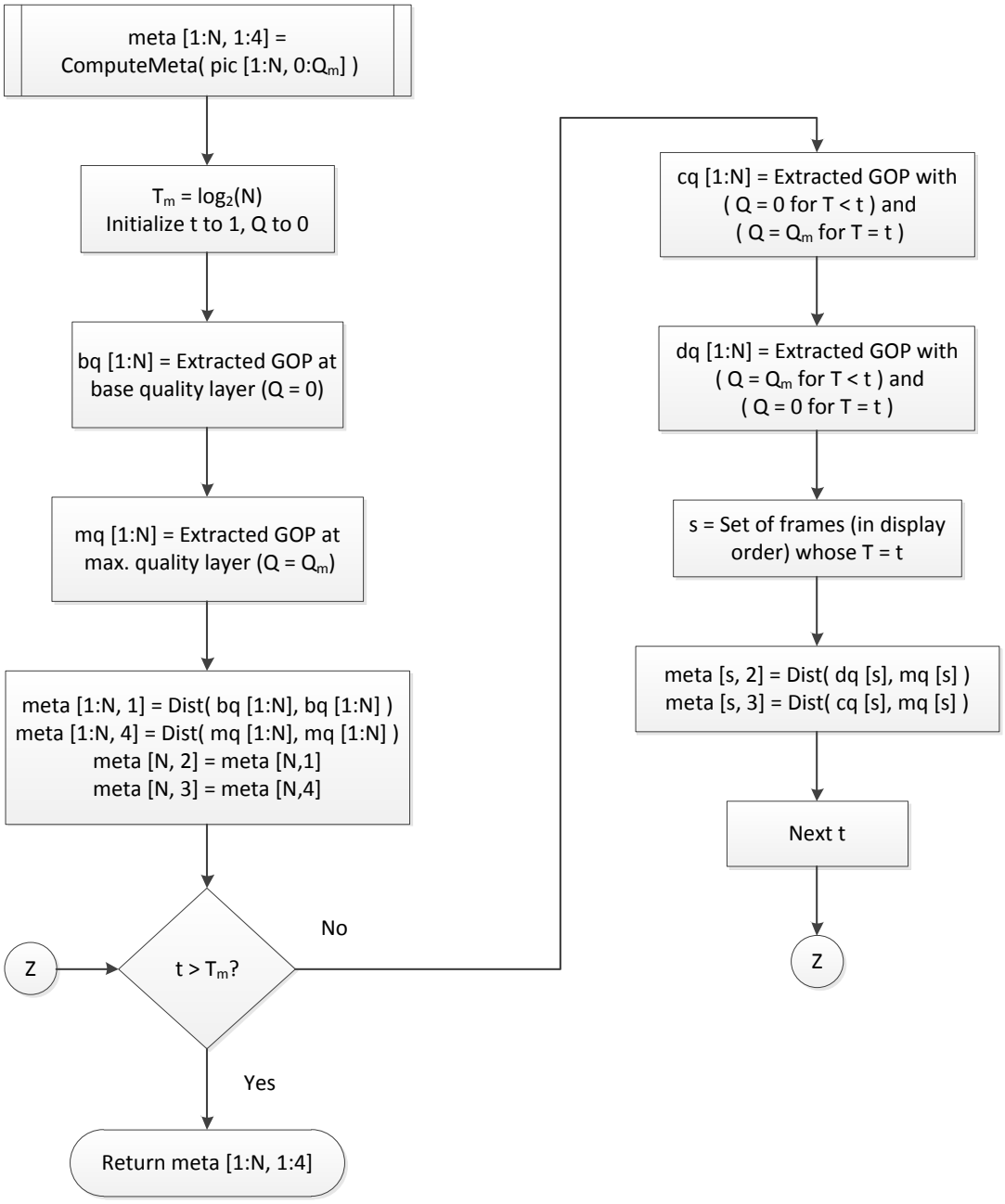


Figure 12: Flowchart for computation of metadata information for MGS quality layers.

Table 1: Distortions represented by the metadata matrix columns for each frame in a GOP.

Column #	Frame's Quality	Frame's Parents' Quality
1	Lowest ($Q = 0$)	Lowest ($Q = 0$)
2	Lowest ($Q = 0$)	Highest ($Q = Q_m$)
3	Highest ($Q = Q_m$)	Lowest ($Q = 0$)
4	Highest ($Q = Q_m$)	Highest ($Q = Q_m$)

Figure 12 summarizes the algorithm for computing the quality metadata information. It takes encoded data of one group of pictures (GOP) as its input argument. The GOP has $Q_m + 1$ quality layers labeled from $Q = 0, 1, 2, \ldots Q_m$. The GOP size is N frames. Hence, the number of temporal layers is $\log_2 N + 1$, labeled from $T = 0, 1, 2, \ldots T_m$. For simplicity, the flow chart depicts the algorithm for a single spatial layer. It is straight forward to extend it to multiple spatial layers. First, the base quality layer ($Q = 0$) of every frame is extracted into a stream named as `bq`. Then, the highest quality level stream ($Q = Q_m$ for every frame in the GOP) is extracted and labeled as `mq`. The number of rows in the metadata matrix equals the number of frames in the GOP. Each row has four columns that represent the distortions as shown in Table 1.

The first and fourth columns can be filled with the distortions of the sequences `bq` and `mq` respectively. However, filling columns two and three require further extraction of partial bitstreams, which is depicted in the loop structure of the flowchart. For every temporal level, two extractions namely `cq` and `dq` are performed, where `dq` represents the stream with the frames at current temporal layer t at the lowest quality ($Q = 0$) and their parents (frames in the GOP with $T < t$) at the highest quality ($Q = Q_m$) and `cq` represents the stream with the frames at current temporal layer t at the highest quality ($Q = Q_m$) and their parents (frames in the GOP with $T < t$)

at the lowest quality ($Q = 0$). For the frame at temporal layer zero, the second column is the same as the first column and the fourth column is the same as the third column. This is because of the key picture concept in SVC, which mandates that the frame at the most important temporal layer be predicted from the lowest quality layer reconstruction of the reference. This is done to avoid the propagation of the drift from one GOP to the next GOP.

The total number of extractions and decodings performed in this computation of metadata equals twice (`cq` and `dq`) the number of temporal layers except zero, added with the two additional extractions at lowest and highest quality layers (`bq` and `mq`), i.e.,

$$\#\text{ of decodings} = 2T_m + 2 = 2(1 + \log_2 N) \tag{8}$$

Hence, the number of decodings increase in the order of $\log_2 N$ as N, the GOP size, increases. This enables the metadata computation to be performed in real-time even at very large GOP sizes such as 64 or 128 frames. This is in contrast with JSVM–QL technique that performs an exhaustive number of decoding operations for every layer, which prevents it from being used in real-time applications.

The above technique can be used only for extracting MGS layers, i.e., they assume that the base quality layer has already been extracted. The extraction of MGS layers does not begin until the base quality layers of all the frames in the current GOP have been extracted. Hence, the extractor also needs a decision making process for RD optimal extraction of the base quality layers. The straight forward technique is to use the temporal layer ID as a guiding factor in extracting base quality layers. Since lower temporal layer pictures are the prediction parents of the higher temporal layer pictures, their base quality layers ($Q = 0$) need to be extracted first. This is followed by the extraction of the base quality layer of the higher temporal layer pictures if the available bandwidth permits such an extraction. Figure 9 shows that their are many pictures at each temporal layer within a GOP. The selection criteria among

them influences the resultant reconstruction video quality. JSVM–Basic technique uses their appearance order in the GOP as the metric in selecting frames belonging to the same temporal layer. This is not a very good metric since it is independent of the video content.

We devise a scheme for the selection of the base quality layers for frames belonging to the same temporal layer. When a frame is not received by the decoder, it copies the previous frame in display order from the decoded picture buffer (DPB). Hence, among the frames belonging to the same temporal layer, we assign least priority to the frame that has the closest similarity to its previous frame in display order since distortion is minimized on replacing this frame with the previous one (frame copy). Hierarchical B-pictures and extracting in increasing temporal order ensure that the previous frame that would be available in the DPB is actually the current frame's parent that occurs before it in display order. We term this parent as the concealment parent. Table 2 shows the concealment parent for the frames in a hierarchical prediction GOP structure such as the one shown in Figure 9. The frame numbers are in display order and Frame # 0 refers to the frame at temporal layer zero of the previous GOP, and it acts as a parent for many frames in this GOP. The first flowchart in Figure 13 computes the distortion (`mse[1:N]`) between each frame and its concealment parent. The second flowchart illustrates how the priority IDs are assigned for the base quality layers of each frame within a GOP. Frames with a lower temporal ID are given a higher priority. For frames within a temporal layer, priority IDs are assigned according to the decreasing order of distortion computed between the base quality layer of the concealment parent and the current frame (`mse[1:N]`). This ensures that the order of extraction of the base quality layers of frames belonging to the same temporal level ensure that distortion is minimized.

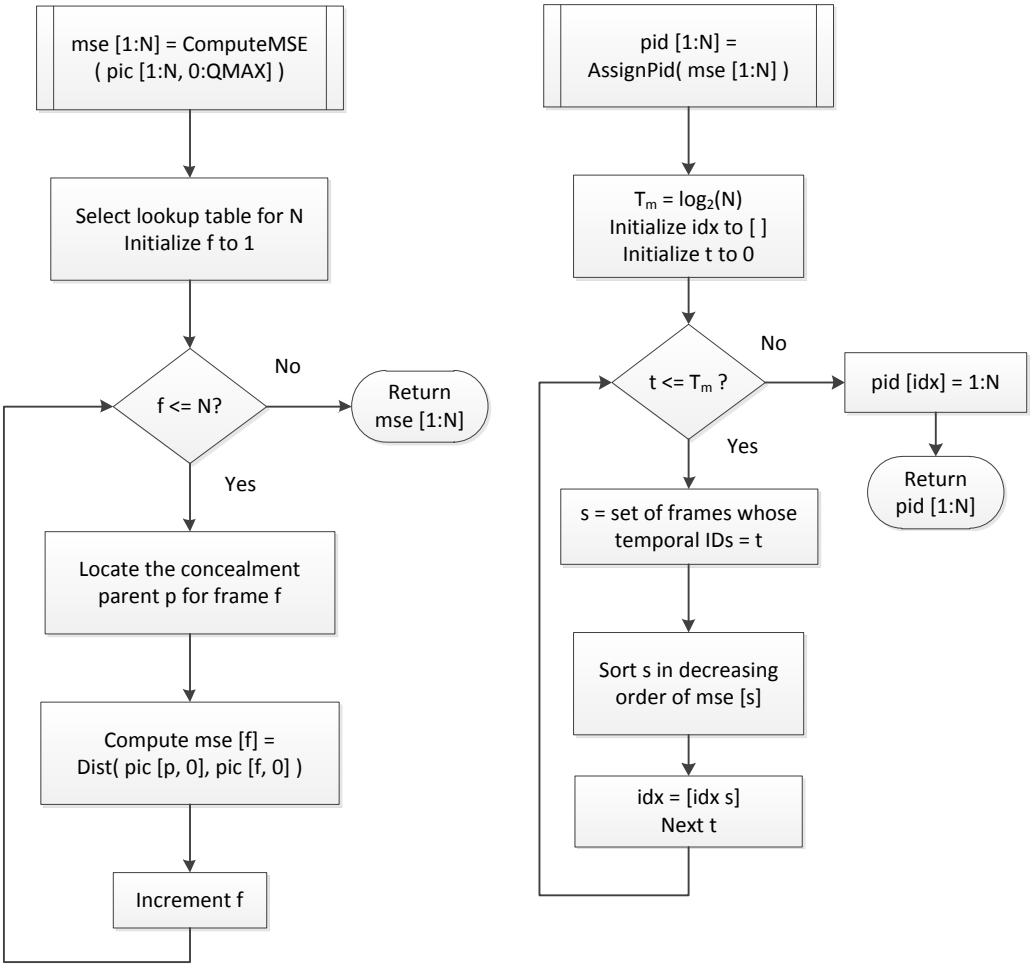


Figure 13: Flowchart for the assignment of priority ID to base quality layers.

Table 2: Frames and their concealment parents in display order for a GOP (8 frames) using a hierarchical prediction structure (Frame # 0 refers to the frame at zero temporal layer in the previous GOP).

Frame # in the GOP	Concealment Parent
1	0
2	0
3	2
4	0
5	4
6	4
7	6
8	0

3.3.2.2 Signaling of Quality Metadata Information

The quality metadata information for the MGS layers and the priority IDs for the base quality layers are transmitted as part of the SVC bitstream either in the *priority_id* field of the NAL unit header or in separate SEI messages. Hence, the extractor can access this information by simply parsing the header and not having to decode the stream.

3.3.2.3 Extraction based on Quality Metadata Information

The extractor node receives the SVC bitstream embedded with the quality metadata information. It measures the available bandwidth between itself and each of its clients or the clients report their available bandwidth to the extractor node. Based on the current channel conditions, it extracts a partial SVC bitstream for each of its clients and transmits it along the respective network paths. The extraction algorithm is illustrated by the flowchart in Figure 14 for a single spatial layer. As the first step, the base quality layers are extracted in the order of decreasing priority IDs that has already been assigned during the post-encoding process. Then, the layer size ratios (`sratio`) are computed. For the base quality layer, it is set to zero since it is not an

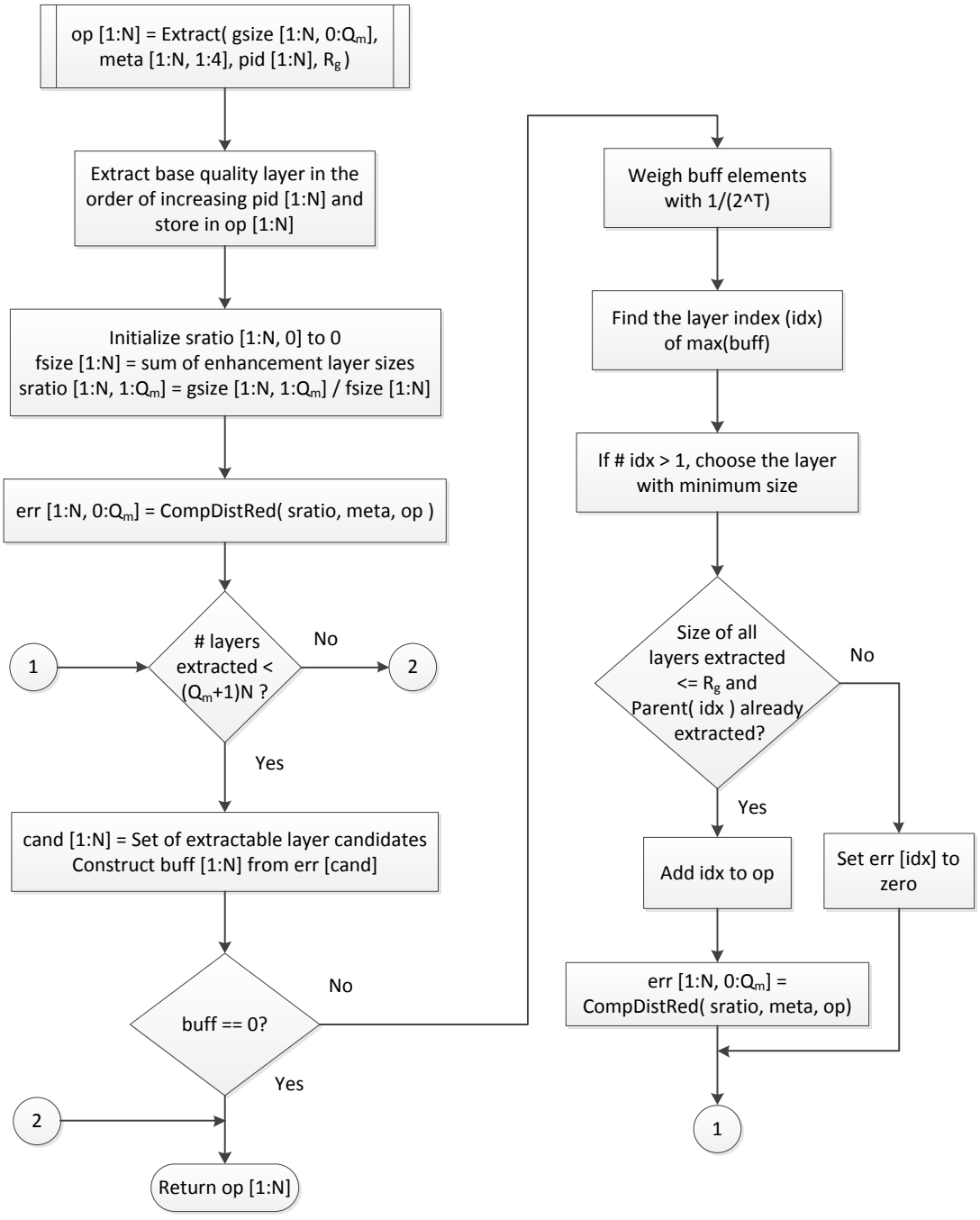


Figure 14: Flowchart for the extraction of layers from an SVC bitstream.

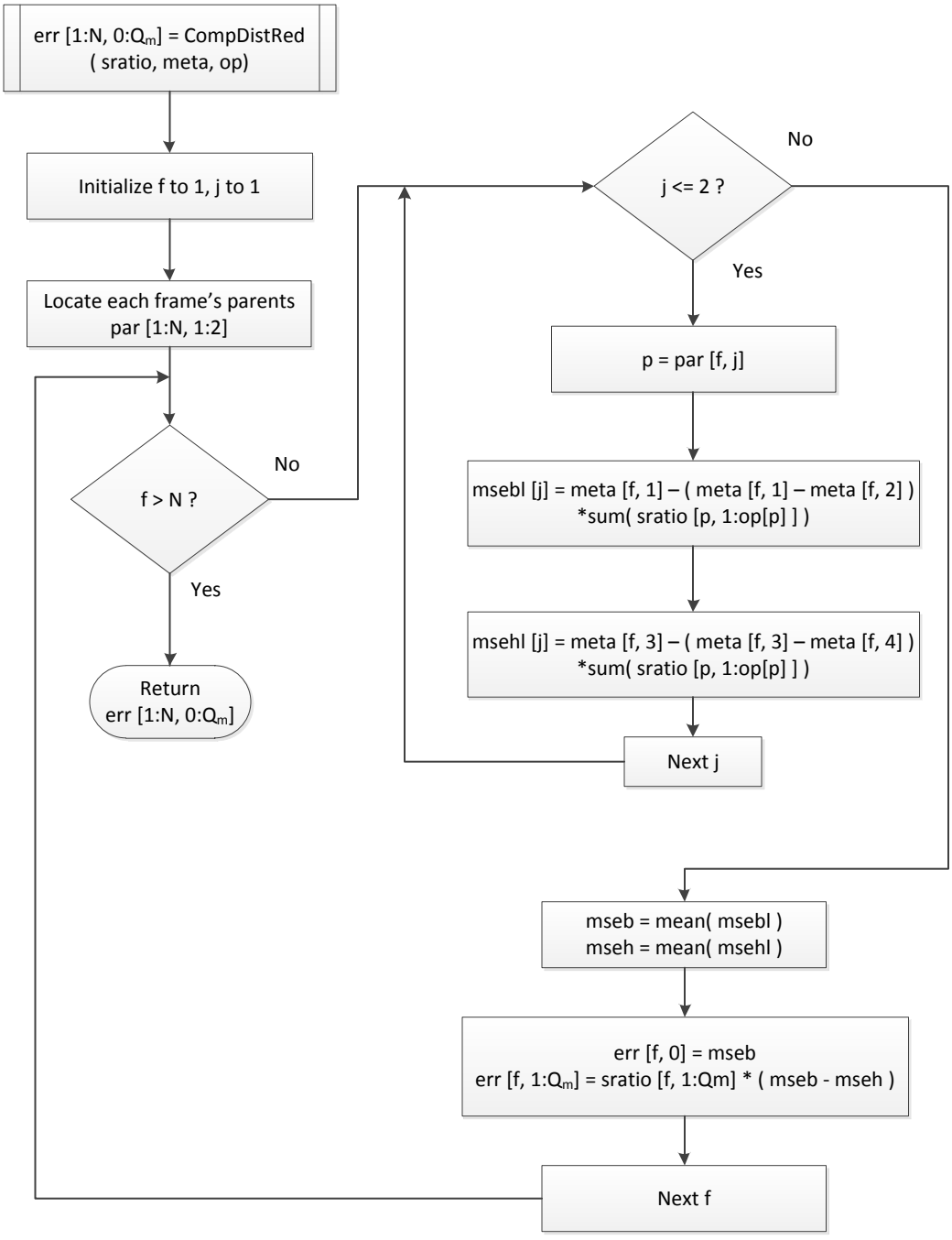


Figure 15: Flowchart for the computation of the GOP's estimated distortion.

MGS layer and has already been extracted. `fsize` computes the sum of all the sizes of the enhancement quality layers (layers with $Q > 0$). `sratio` assigns the size ratio for each quality layer ranging from $Q = 1, 2, \ldots Q_m$ for all the N frames in the GOP.

The distortion of the various quality layers in the current GOP is computed using the function `CompDistRed()`, which is explained in detail in the flowchart in Figure 15. This function takes the layer size ratios (`sratio`), metadata (`meta`) matrix, the list of layers already extracted in the current GOP (`op`) as its input and returns the `err` matrix, whose size (rows $\times$ columns) is equal to the number of layers (temporal $\times$ quality) in the GOP. The first column in this matrix contains the estimated distortions of the base quality layer of each frame in the GOP. The next column contains the estimated reduction in distortion obtained for each frame when their MGS quality layer at $Q = 1$ is extracted. Similarly, the other columns contain the estimated reduction in distortion obtained as the higher MGS quality layers are extracted. As Figure 15 shows, the computation of `err` starts with the finding of parents of the current frame in the GOP. Depending on the number of quality layers already extracted for each parent, the distortions of the base quality layer and the highest quality layer of the current frame is estimated.

We saw already that the first two columns of `meta` give the distortion of the base layer of the current frame when it is predicted from the lowest and the highest quality reconstruction of its parents. Hence, depending on the number of quality layers that has already been extracted for each parent `j`, the estimated distortion of the base layer of the current frame `msebl[j]` is computed as the distortion obtained from the base layer reconstruction of each parent (`meta[f,1]`) reduced by the number of additional MGS layers extracted for each parent. The reduction in distortion is computed by subtracting the first and second columns of `meta`. This difference is scaled down by the size ratio of all the MGS layers extracted for each parent. The scaling is necessary since columns 1 and 2 of `meta` represent the distortions of the base layer of the current

frame predicted from the lowest and highest quality reconstructions of its parents and in the current scenario, not all MGS quality layers would have been extracted for the parent.

Next, the estimated distortion of the highest quality layer ($Q = Q_m$) is computed for the current frame `f` in a manner similar to the estimation of base layer distortion. Columns 3 and 4 of the `meta` matrix represent the distortion of the highest quality layer of the current frame `f` predicted from the lowest and highest quality reconstructions of its parents. Hence, the distortion of the highest quality layer is estimated from the distortion of that layer predicted from the lowest quality reconstruction of its parents (`meta[f,3]`) reduced by the distortion reduction obtained due to the extraction of additional MGS layers for each parent. This amounts to subtracting columns three and four of `meta` and scaling down the difference by the size ratio of all the MGS layers extracted for each parent. This process is repeated for each parent. The final estimated distortion for the base layer and the highest quality layer of the current frame `f`, represented as `mseb` and `mseh`, is computed as the average of the respective distortions obtained from each parent.

Finally, the row in `err` matrix corresponding to the current frame `f` is filled. The first column is filled with `mseb`, i.e., the distortion of the base layer of the current frame. The rest of the columns are filled with the reduction in distortion obtained on extracting additional MGS layers ($Q > 0$). Since the only available estimated distortion for the current frame is `mseh` and `mseb`, the reduction in distortion for all the in between MGS layers is obtained by subtracting the difference between the lowest and highest quality layer distortions (`mseb`$-$`mseh`) and distributing this difference among the in between MGS layers in proportion of their sizes. The entire process is repeated for all the N frames in the GOP.

Coming back to Figure 14, the extraction process begins with the selection of available candidates for extraction (`cand`), which is based on the candidate layers

satisfying the dependency relations with the layers that have already been extracted. A buffer for comparison (`buff`) is created. The buffer is filled with data from the corresponding columns in `err`, which represents the reduction in distortions obtained by decoding each of the candidates available for extraction. The buffer elements are weighed with a factor corresponding to their temporal IDs. This is done to give more importance to lower temporal layers as they are the prediction parents of the frames that belong to higher temporal layers. The index of the layer (`idx`) that maximizes the distortion reduction is identified. If more than one layer maximizes the distortion reduction, then the layer with minimum size is chosen. Now, the dependency checks are verified for this layer (`idx`) and its size is compared with the available bit budget. Once these conditions are satisfied, the layer is added to the extraction list (`op`). Since, the addition of this layer will influence the distortion of this frame and of all its children at higher temporal layers, the current estimated distortion for all frames in the GOP is updated again using `CompDistRed()`. If the size of `idx` is too big for the available bandwidth, then the corresponding position in the `err` matrix is set to zero and the process is repeated so that we can evaluate the next best candidate from `cand`. The entire extraction process is repeated until all the layers in the bitstream have been extracted or the available bit budget (i.e., the bandwidth in the channel) has been reached. The list of layers to be extracted (`op`) is returned from this function. Finally, the layers are extracted from the bitstream according to `op`.

When computing the distortion measure, such as mean square error, we use the SVC stream extracted at the highest spatial, quality and temporal layer as the reference. This allows the computation of quality contribution of layers at any part in the distribution chain. This approach is correct since we are interested in the distortion difference obtained on decoding every additional layer. When all the layers are extracted, the distortion is the same as that of the reference and hence the distortion difference is zero. When a subset of layers are extracted, there exists a distortion

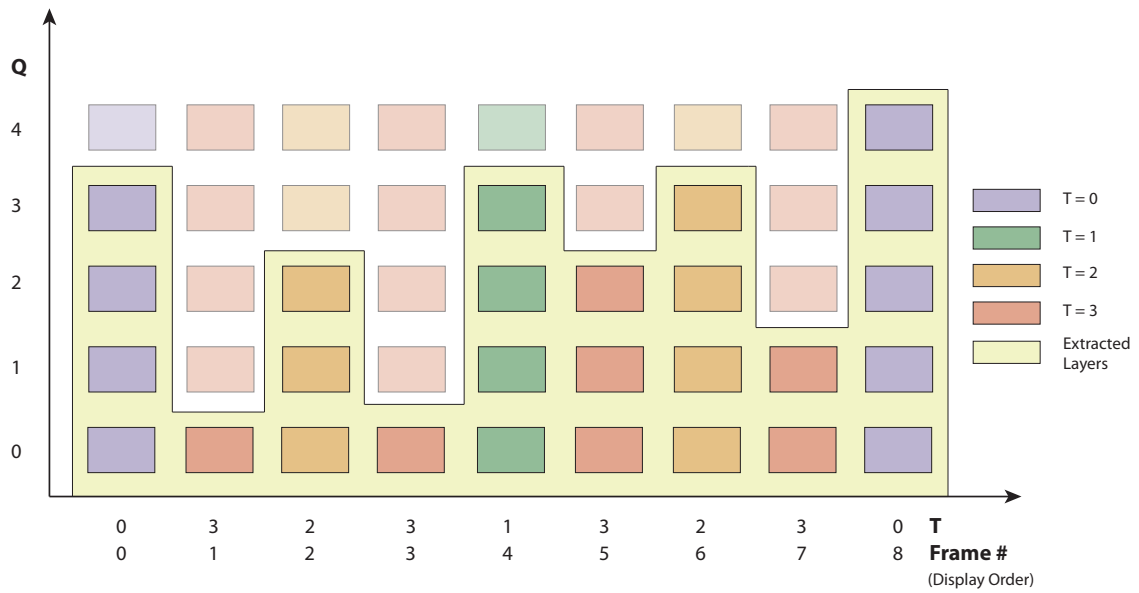


Figure 16: Typical order of layer extraction using the proposed technique for a GOP size of 8 ($T = 0, 1, 2, 3;\ Q = 0, 1, 2, 3, 4;\ D = 0$).

difference that is greater than zero. Since video is captured in YUV 4:2:0 format usually, we use weighted average of the Y, U and V components in computing the distortions.

$$\mathrm{PSNR} = \frac{1}{6}(4\,\mathrm{PSNR_Y} + \mathrm{PSNR_U} + \mathrm{PSNR_V}) \tag{9}$$

As we have seen above, our algorithm addresses all the issues present in the previous techniques. A typical extraction of quality and temporal layers for a single spatial layer is shown in Figure 16. In this example, the GOP size is 8 frames and has a dyadic hierarchical prediction structure as previously shown in Figure 9. There are four enhancement quality layers each containing a certain number of transform coefficients. From the figure, we can see that for certain frames all the quality layers have been extracted where as for a few others only the base quality layer have been extracted depending on the quality contribution of each layer to the reconstructed video and the available bit budget.

3.4 Experiments and Results

In this section, our proposed extraction algorithm is validated through various experiments. The cases of MGS quality layer extraction and base quality layer extraction are treated separately since they involve different decision making approaches. For MGS quality layer extraction, our technique is compared with two current state-of-the-art techniques, namely JSVM–Basic and JSVM–QL. For base quality layer extraction, our technique is compared with JSVM–Basic since the performance of JSVM–QL is identical to that of JSVM–Basic while extracting base quality layers. On an average, our algorithm achieves a gain in video quality of about 1.5 dB over JSVM–Basic and a gain of about 0.5 dB over JSVM–QL. The maximum gain is about 4.0 dB when compared to JSVM–Basic and about 1.5 dB when compared to JSVM–QL. The time required for computing the metadata information during the post-encoding phase is 73% lesser for the proposed technique when compared with JSVM–QL. The sequences used and the encoding parameters are described in the first section. This is followed by the experiments and results for MGS quality layer extraction, base quality layer extraction and metadata computation time. Then, a snapshot of our algorithm's performance is demonstrated. This is followed by comparisons between estimated and actual distortions. Finally, a few sample frames are shown that illustrate the superiority of our technique.

3.4.1 Video Sequence Database

The sequences used in our experiments are the standard test sequences used in the video research community and standardization bodies for testing video compression algorithms and encoder optimizations. Our database consists of nine sequences divided into three sets based on their spatial resolutions. There are five sequences at 1280×720 (720p) resolution in SET 1 , two at 352×288 (CIF) resolution in SET 2 and two at 176×144 (QCIF) resolution in SET 3. The sequence names, properties and

SVC encoding parameters are described in Table 3. Our database is representative of a wide variety of characteristics. Aspen sequence has extensive spatial details with little motion between frames where as the Touchdown sequence has quick motion with lesser spatial details. The Field rush sequence has random motion covering the entire spatial area. In the Red Kayak sequence, the motion is limited to certain areas where the person, kayak and the waves are located. Different spatial resolutions help us test the effectiveness of our algorithm over a wide spectrum of sequence characteristics and bitrates. The scan type is progressive in YUV 4:2:0 format. The frame rate is 25 fps for all the sets. The GOP structure uses hierarchical B-pictures, as shown is Figure 9. With a GOP size of 8 frames, the number of GOPs used is 50 for SET 1. Due to a difficulty in obtaining 50 GOPs for SET 2 and SET 3, the number of GOPs are limited to 30 for these SETs. All the sequences include an additional frame in the beginning, which is encoded as an IDR picture. All the sequences are encoded using the JSVM SVC encoder [70], which is the reference software issued by ITU–T. The quantization parameters (QP) used for encoding are also shown in Table 3. Since the GOP size used is 8, there are four temporal layers in all the sets. For simplicity, only one spatial layer with six quality layers (including the base quality layer) is used. Sample frames from all the sequences are shown in Figure 17. The encoded bitrate of each layer for all the sequences in SET 1 is shown in Table 4 and in Table 5 for SET 2 and SET 3. From the table, we can see that our database covers an extremely wide range of bitrates, from 100 kb/s to 6000 kb/s.

3.4.2 Video Quality: MGS Quality Layer Extraction

In this section, we describe the experiments conducted and the results obtained for the extraction of MGS quality layers. Hence, the range of the available bandwidth used for experiments in this section are chosen in a way such that that extraction of the base quality layers of all the frames in the GOP is guaranteed. In such cases, the

Table 3: Test sequences' characteristics and encoding parameters.

Parameter	SET 1	SET 2	SET 3
# of sequences	5	2	2
Sequence names	Aspen, Rush hour, Field rush, Red kayak, Touchdown	Mobile, City	Carphone, Coastguard
Spatial resolution	1280×720 (720p)	352×288 (CIF)	176×144 (QCIF)
Scan type	Progressive	Progressive	Progressive
YUV format	4:2:0	4:2:0	4:2:0
Frame rate	25 fps	25 fps	25 fps
# of frames	401	241	241
Duration	16.04 s	9.64 s	9.64 s
GOP size (N)	8 frames	8 frames	8 frames
Sequence structure	IDR–{B3-B2-B3-B1-B3-B2-B3-P0}×50	IDR–{B3-B2-B3-B1-B3-B2-B3-P0}×30	IDR–{B3-B2-B3-B1-B3-B2-B3-P0}×30
Base layer QP	40	30	30
MGS layer QP	30	20	20
# of Temporal layers	4 ($T = 0, 1, 2, 3$)	4 ($T = 0, 1, 2, 3$)	4 ($T = 0, 1, 2, 3$)
# of Quality layers	6 ($Q = 0, 1, \ldots 5$)	6 ($Q = 0, 1, \ldots 5$)	6 ($Q = 0, 1, \ldots 5$)
# of Spatial layers	1 ($D = 0$)	1 ($D = 0$)	1 ($D = 0$)

Table 4: Bitrates (kb/s) of the SVC encoded sequences in SET 1 (720p).

Layers (D, T, Q)	Aspen	Rush Hour	Field Rush	Touchdown	Red Kayak
(0,0,0)	577.10	153.50	725.70	222.70	383.10
(0,1,0)	736.10	213.70	966.00	318.90	600.40
(0,2,0)	912.40	285.00	1206.20	428.70	896.70
(0,3,0)	1056.20	356.60	1393.90	547.80	1245.70
(0,0,1)	1514.50	761.00	1794.00	937.20	1111.40
(0,0,2)	2047.00	860.70	2340.00	1164.00	1373.30
(0,0,3)	2368.00	881.70	2641.00	1268.10	1495.20
(0,0,4)	2557.00	890.50	2821.00	1318.40	1572.90
(0,0,5)	2633.00	894.80	2886.00	1334.10	1598.50
(0,1,1)	2073.00	1007.30	2561.00	1245.30	1847.00
(0,1,2)	2766.00	1128.50	3278.00	1518.80	2246.00
(0,1,3)	3169.00	1161.40	3655.00	1648.00	2431.00
(0,1,4)	3395.00	1180.90	3868.00	1714.00	2548.00
(0,1,5)	3484.00	1195.90	3944.00	1740.00	2587.00
(0,2,1)	2770.00	1325.60	3453.00	1613.70	2945.00
(0,2,2)	3626.00	1477.10	4331.00	1941.00	3538.00
(0,2,3)	4111.00	1532.40	4779.00	2104.00	3814.00
(0,2,4)	4376.00	1573.60	5026.00	2194.00	3988.00
(0,2,5)	4487.00	1610.20	5122.00	2240.00	4050.00
(0,3,1)	3473.00	1709.00	4294.00	2036.00	4418.00
(0,3,2)	4471.00	1908.00	5304.00	2428.00	5276.00
(0,3,3)	5031.00	2004.00	5823.00	2637.00	5687.00
(0,3,4)	5340.00	2086.00	6118.00	2767.00	5940.00
(0,3,5)	5488.00	2163.00	6258.00	2851.00	6036.00

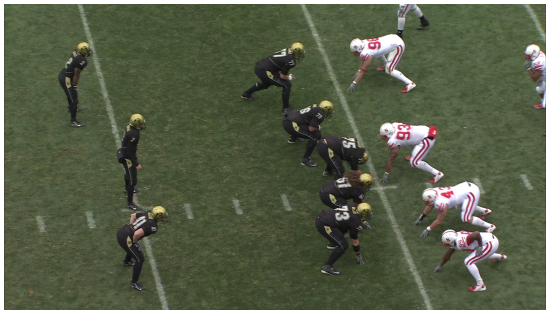
Table 5: Bitrates (kb/s) of the SVC encoded sequences in SET 2 (CIF) and SET 3 (QCIF).

Layers (D, T, Q)	Mobile	City	Carphone	Coastguard
(0,0,0)	406.80	176.90	47.60	74.80
(0,1,0)	510.80	202.10	64.50	97.20
(0,2,0)	634.20	231.50	86.20	114.50
(0,3,0)	753.00	266.50	110.40	125.20
(0,0,1)	585.80	340.30	87.10	114.00
(0,0,2)	753.30	476.70	115.70	147.90
(0,0,3)	908.30	590.40	139.00	176.80
(0,0,4)	1050.90	687.40	159.60	205.70
(0,0,5)	1198.10	766.70	178.40	232.70
(0,1,1)	798.50	420.60	124.70	163.30
(0,1,2)	1058.00	588.90	163.40	217.80
(0,1,3)	1294.10	727.00	194.90	262.10
(0,1,4)	1505.90	842.50	223.10	307.60
(0,1,5)	1711.00	932.40	248.80	344.90
(0,2,1)	1083.00	517.20	174.80	216.90
(0,2,2)	1470.60	715.10	226.70	296.70
(0,2,3)	1818.00	875.80	268.80	359.20
(0,2,4)	2120.00	1008.40	306.70	424.20
(0,2,5)	2396.00	1107.80	341.30	471.80
(0,3,1)	1450.80	653.30	241.50	267.30
(0,3,2)	2029.00	888.80	312.30	369.60
(0,3,3)	2545.00	1080.30	369.70	449.20
(0,3,4)	2970.00	1235.30	421.90	531.30
(0,3,5)	3346.00	1350.00	468.90	589.10

(a) Aspen

(b) Rush hour

(c) Field rush

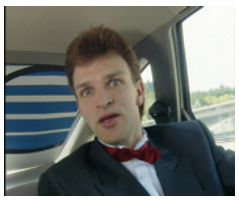


(d) Touchdown

(e) Red kayak

(f) Mobile

(g) City



(h) Carphone

(i) Coastguard

Figure 17: (a) – (e): Sample frames from SET 1 (720p), (f) – (g): Sample frames from SET 2 (CIF), (h) – (i): Sample frames from SET 3 (QCIF).

extracted sequence operates at full frame rate and the reconstructed video quality depends on an RD-optimal extraction of the MGS quality layers. This enables us to test the power of our algorithm in performing such an RD-optimal extraction. Two state-of-the-art extraction techniques namely JSVM–Basic and JSVM–QL [60] have been used for comparison purposes. JSVM–Basic performs a content-independent extraction and JSVM–QL extraction is based on quality layers concept and is defined only for MGS quality layers. The performance measure by which all the algorithms are compared is that of reconstructed video quality, which is measured through a full-reference metric, such as PSNR. For each available bandwidth value, the SVC bitstream is extracted according to our proposed technique and also with JSVM–Basic and JSVM–QL techniques. These extracted sequences are then decoded and their reconstructed video quality (PSNR) is compared.

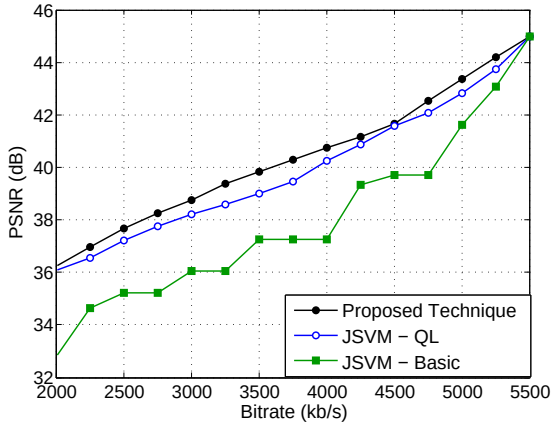
In Figure 18, the variation of video quality (in terms of PSNR measured in dB) with respect to the available bandwidth in the channel is plotted for all the five sequences (Aspen, Rush hour, Field rush, Touchdown, Red kayak) in SET 1 (720p). As it can be seen from the plots, the proposed technique always offers better reconstructed video quality than JSVM–QL and JSVM–Basic. This is because our technique estimates the distortion of the current GOP accurately and updates it every time after a layer has been extracted. This leads to a better knowledge of the quality contribution of the remaining layers in the GOP and hence, RD optimal decisions are made during extraction. JSVM–QL computes the impact of the refinement quality layers assuming that all the lower level quality planes are included. This results in inaccurate calculations of distortion reduction, which reduces the RD optimality of the technique. Since JSVM–Basic makes extraction decisions simply on the basis of quality and temporal layer ID in a content-independent fashion, its performance is the lowest. It can also be noticed that though JSVM–QL performs better than JSVM–Basic in most cases, JSVM–Basic has higher reconstructed video

Table 6: Mean and Max. increase in PSNR (dB) over JSVM–QL & JSVM–Basic for the extraction of MGS layers of SET 1 (720p) sequences.

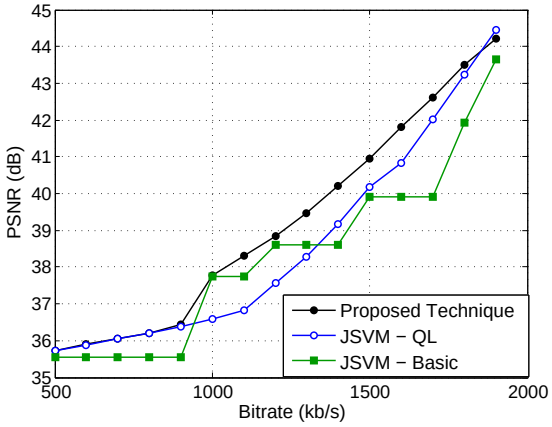
Sequence	Mean increase over JSVM–QL (dB)	Max. increase over JSVM–QL (dB)	Mean increase over JSVM–Basic (dB)	Max. increase over JSVM–Basic (dB)
Aspen	0.46	0.85	2.39	3.49
Rush hour	0.58	1.50	0.91	2.70
Field rush	0.35	0.69	1.28	2.51
Touchdown	0.42	0.84	1.12	1.97
Red kayak	0.82	1.41	2.22	4.09

quality than JSVM–QL for the bandwidth range of 900 kb/s – 1300 kb/s of the Rush hour sequence. This is because of the fact that the content-independent extraction strategy adopted by JSVM–Basic turns out to be close to an RD optimal extraction in this specific case. However, even in that scenario, our algorithm performs better than both JSVM–Basic and JSVM–QL. Table 6 summarizes the results of the plots in Figure 18. For all the sequences in SET 1, it shows the mean and maximum increase in video quality (PSNR) obtained over JSVM–QL and JSVM–Basic when extraction is performed using the proposed algorithm. The average is computed over the entire bandwidth range for each sequence. From the table, we see that the mean increase in PSNR is about 0.5 dB (averaged over all sequences) when compared to JSVM–QL, with a highest increase of 1.5 dB for the Rush hour sequence. The mean increase over JSVM–Basic is about 1.6 dB, with a highest increase of 4.09 dB for the Red kayak sequence.

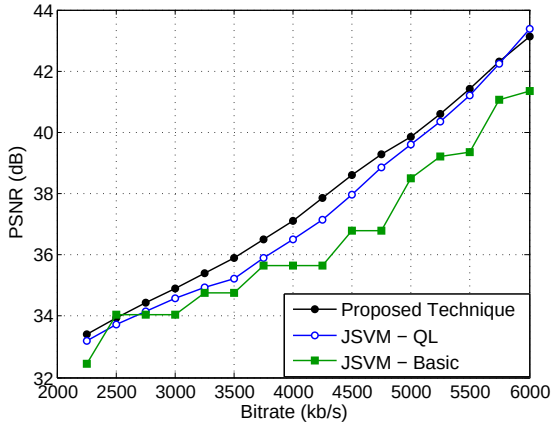
The above experiment of evaluating the proposed algorithm's performance in performing an RD-optimal extraction of MGS quality layers is also conducted on the CIF sized sequences in SET 2 and the QCIF sized sequences in SET 3. This helps us in



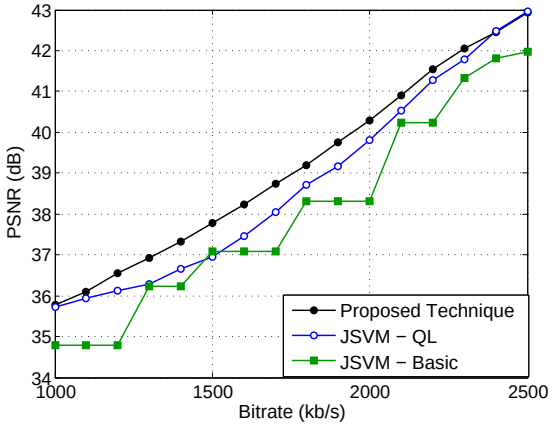
(a) Aspen



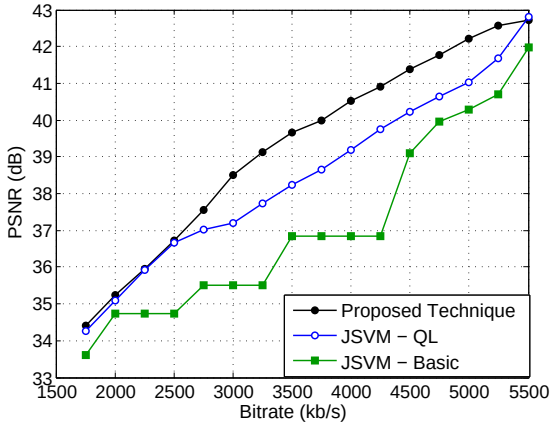
(b) Rush hour



(c) Field rush



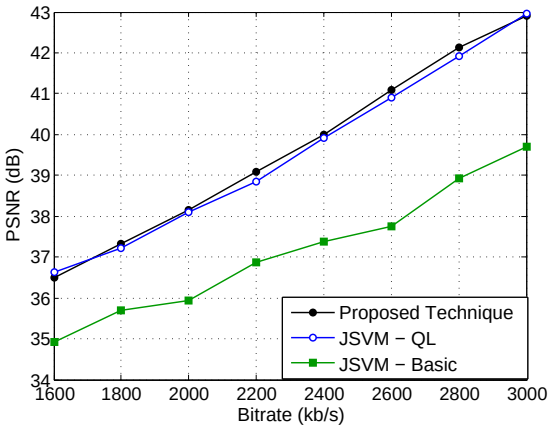
(d) Touchdown



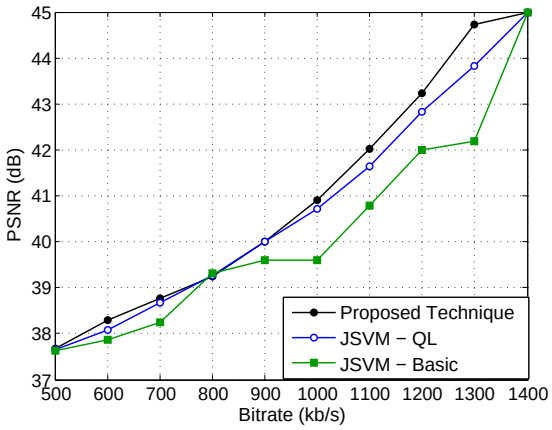
(e) Red kayak

Figure 18: Video quality (PSNR) vs. bitrate (available bandwidth) for SET 1 (720p) sequences for MGS layers extraction.

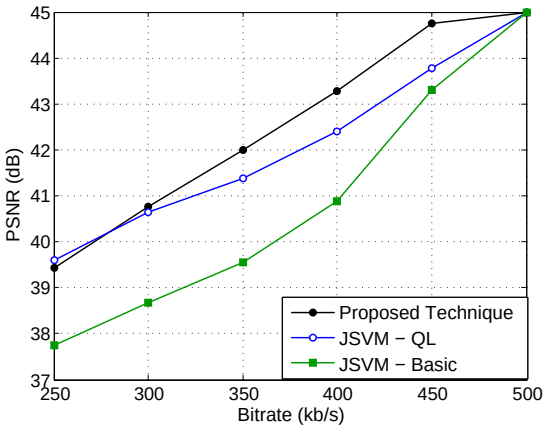
validating the algorithm in a wide variety of spatial resolutions and bitrates. Figure 19 shows the variation of video quality (PSNR) with respect to the available bandwidth in the channel for the Mobile and City sequences in SET 2 (CIF) and the Carphone and Coastguard sequences in SET 3 (QCIF). Even at reduced spatial dimensions, we see that our algorithm provides a higher video quality than JSVM–Basic and JSVM–QL. We also notice that at these lower resolutions, JSVM–QL always performs better than JSVM–Basic. Table 7 summarizes the results of the plots in Figure 19. For all the sequences in SET 2 and SET 3, it shows the mean and maximum increase in video quality (PSNR) obtained over JSVM–QL and JSVM–Basic when extraction is performed using the proposed algorithm. As in SET 1, the average is computed over the entire bandwidth range for each sequence. From the table, we see that the mean increase in PSNR is about 0.3 dB (averaged over all sequences) when compared to JSVM–QL, with a highest increase of 0.98 dB for the Carphone sequence. The mean increase over JSVM–Basic is about 1.6 dB, with a highest increase of 3.44 dB for the Mobile sequence. The mean improvement in PSNR over JSVM–QL is 0.2 dB less for SET 2 and SET 3 when compared to SET 1. This is attributed to the lesser number of sequences and smaller total bitrates in SET 2 and SET 3 over which the mean is computed and the number of frames in each sequence (241 in SET 2 and 3, and 401 in SET 1). Also at lower spatial resolutions, the number of NAL units at each quality layer is usually one. This leads to a binary decision process, i.e., whether that quality layer can be sent or not. But at higher spatial resolutions, due to a larger number of macroblocks, each quality layer is split and packed into mulitple NAL units, which gives us the additional flexibility of transmitting only those NAL units of a quality layer whose sizes are within the available bit budget. This leads to an overall increase in reconstructed video quality at higher spatial resolutions.



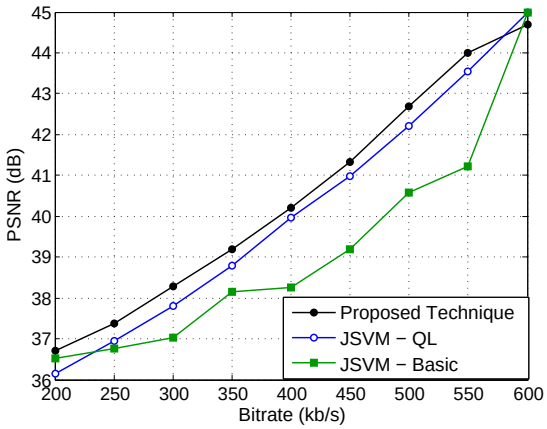
(a) Mobile



(b) City



(c) Carphone



(d) Coastguard

Figure 19: Video quality (PSNR) vs. bitrate (available bandwidth) for SET 2 (CIF) and SET 3 (QCIF) sequences for MGS layers extraction.

Table 7: Mean and Max. increase in PSNR (dB) over JSVM–QL & JSVM–Basic for the extraction of MGS layers of SET 2 (CIF) and SET 3 (QCIF) sequences.

Sequence	Mean increase over JSVM–QL (dB)	Max. increase over JSVM–QL (dB)	Mean increase over JSVM–Basic (dB)	Max. increase over JSVM–Basic (dB)
Mobile	0.09	0.24	2.50	3.44
City	0.22	0.90	0.77	2.55
Carphone	0.41	0.98	1.68	2.46
Coastguard	0.35	0.56	1.31	2.79

3.4.3 Video Quality: Base Quality Layer Extraction

In this section, we describe the experiments and results obtained for the extraction of base quality layers. In the previous section, the focus was on the extraction of MGS layers. MGS quality layer extraction does not begin until all the base quality layers of all the frames in the GOP have been extracted. During moderate to high bandwidth conditions, there is usually sufficient bandwidth in the channel to allow the extraction of all the base quality layers. Hence, this is followed by an RD optimal extraction of MGS layers. When the available bandwidth in the channel drops to low levels, there may not be sufficient bandwidth to extract all of the base quality layers. So the extractor needs to perform an RD optimal extraction of the base quality layers within the low available bandwidth. This would result in skipping of the base quality layer of one or more lesser important frames in the GOP, thus leading to missing frames at the decoder, which may use concealment options like frame copy to compensate for the unreceived frame. The experimentation conditions are similar to the ones described in the previous section. Reconstructed video quality measured through PSNR is taken as an indicator of performance. Extraction is performed using our base quality layer extraction algorithm based on priority ID assignment and using

JSVM–Basic. The decoded video quality is compared for both techniques at various available bandwidth values. Since JSVM–QL is defined only for the extraction of MGS layers, its performance in the extraction of base quality layers is identical to that of JSVM–Basic and hence, JSVM–QL is not dealt as a separate technique for comparison of base quality layer extraction.

The experiments for base quality layer extraction are conducted with SET 1 only. SET 2 and SET 3 are avoided since their bitrates from Table 5 suggests that the bandwidth required for complete base quality layer extraction ($T = 3, Q = 0, D = 0$) is 266.50 kb/s for City, 110.40 kb/s for Carphone and 125.20 kb/s for Coastguard. Except for the Mobile sequence, whose base layer bitrate is 753.00 kb/s, all the other sequences' base quality layers can be extracted completely even at very low bandwidths of a few hundred kb/s and hence, they do not represent an interesting case for RD optimal extraction of the base quality layer. Figure 20 shows the variation of video quality (PSNR) with respect to the available bandwidth in the channel for the five sequences in SET 1 (720p). The figure shows that our proposed technique for base quality layer extraction always performs better than JSVM–Basic. This is because our technique is RD optimal and content-dependent. It is based on priority ID assignment to each base quality layer in the GOP based on the ease of reconstruction from its concealment parent, in the event if the frame is skipped. This leads to informed decisions in dropping those base quality layers of frames that can be concealed very easily with minimum distortion. The lower performance of JSVM–Basic technique is due to the fact that its decisions are solely based on temporal ID and is content-independent. The 'step' nature of the curves for JSVM–Basic is due to the fact that for consecutive available bandwidth, the extracted stream is the same (i.e., corresponding to a reduced frame rate). For example, in the Aspen sequence of Figure 20, for bandwidth 600 and 700 kb/s, the extracted stream corresponds to temporal layer 1 (frame rate: 6.25 fps), and for bandwidth 800 and 900 kb/s, the extracted stream

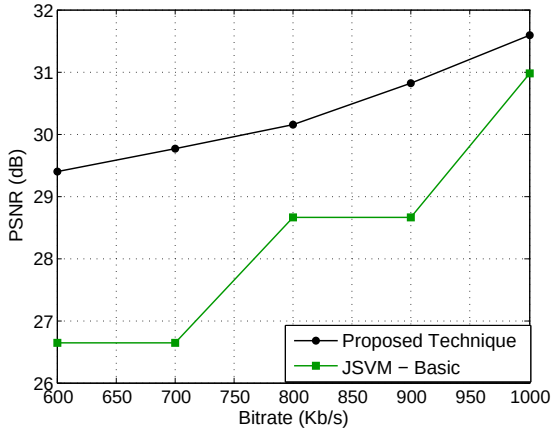
Table 8: Mean and maximum increase in PSNR (dB) over JSVM–Basic for the extraction of base quality layers of SET 1 (720p) sequences.

Sequence	Mean increase over JSVM–Basic (dB)	Max. increase over JSVM–Basic (dB)
Aspen	2.03	3.12
Rush hour	0.99	1.99
Field rush	1.74	3.04
Touchdown	1.31	2.21
Red kayak	1.20	2.38

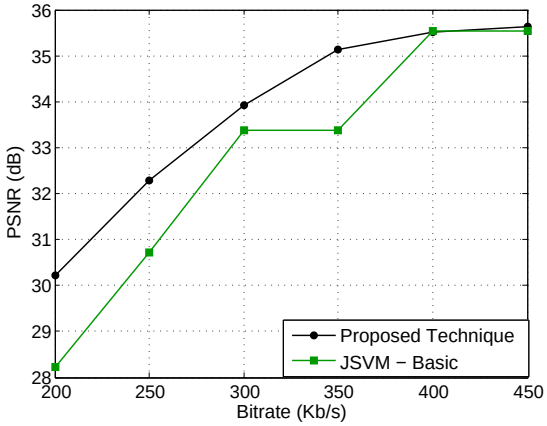
corresponds to temporal layer 2 (frame rate: 12.50 fps). Our algorithm is able to take advantage of the full available bandwidth each time, thus generating a smooth RD curve. This is because frame skipping is not done on a temporal level-by-level basis, but is handled in a more fine grained manner (frame-by-frame basis). Table 8 summarizes the results for all the sequences in SET 1. It shows the mean increase in PSNR achieved by the proposed technique is about 1.5 dB over JSVM–Basic, with a maximum increase of 3.12 dB for the Aspen sequence. The mean increase is similar to the improvement obtained in the previous section for MGS quality layer extraction. This shows that consistent performance improvement is achieved by using our algorithm for both MGS and base quality layer extraction when compared to other extraction techniques.

3.4.4 Metadata Computation Time

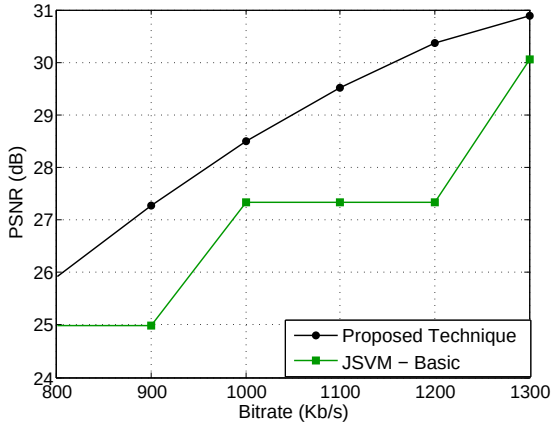
In this section, we evaluate the time required for computing the metadata needed for extraction using the proposed technique and compare it with the time required for the computation of the quality layer information needed for extraction using JSVM–QL. Since JSVM–Basic performs extraction independent of metadata, it is not a part of



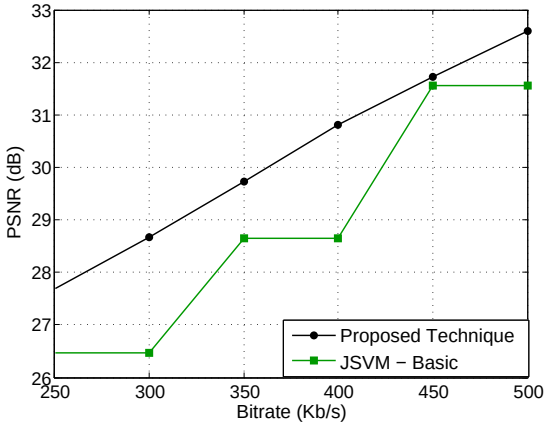
(a) Aspen



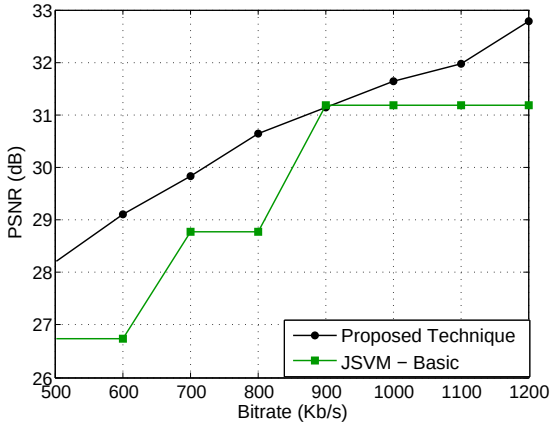
(b) Rush hour



(c) Field rush



(d) Touchdown



(e) Red kayak

Figure 20: Video quality (PSNR) vs. bitrate (available bandwidth) for SET 1 (720p) sequences for base quality layer extraction.

Table 9: Metadata computation time (seconds) for JSVM–QL and the proposed technique for SET 1 (720p), SET 2 (CIF) & SET 3 (QCIF) sequences.

SET #	Sequence	JSVM–QL (seconds)	Proposed Technique (seconds)	Reduction (%)
SET 1	Aspen	4048.00	1090.20	73.07
	Rush hour	4005.00	1080.96	73.00
	Field rush	4097.00	1103.13	73.07
	Touchdown	4036.00	1083.96	73.14
	Red kayak	3991.00	1070.24	73.18
SET 2	Mobile	232.88	64.18	72.44
	City	220.83	60.08	72.79
SET 3	Carphone	47.29	13.23	72.02
	Coastguard	48.02	13.49	71.90

this comparison. Table 9 shows the time (reported in seconds by the `time` command) taken for computing this metadata information by a system with a memory of 4GB RAM and running on an Intel Xeon quad-core processor. The operating system used is Ubuntu 10.04 LTS (Linux). The metadata required for the proposed technique and JSVM–QL is computed one after the other for each sequence in SET 1, SET 2 and SET 3. No other application or user-level process is run while the metadata is being computed. This helped maintain the CPU usage for the metadata computation process at a constant rate of 99% (reported by the `time` command).

As the table shows, the time required to compute the metadata needed for extraction using the proposed technique is 73% lesser, averaged across all sequences, than the time required for computing the quality layer metadata needed for extraction using JSVM–QL. This huge reduction in metadata computation time along with the improvements in video quality make our technique a more preferred candidate

than JSVM–QL for use in real-time streaming applications. The reduction in metadata computation time is due to the limited number of decodings (of the order of $\log_2 N$, as shown in Equation (8)) that are required by our technique. We perform decodings only at the lowest and highest quality layers of each temporal layer that has been predicted from the lowest and highest quality layers of their parents. For the in between MGS quality layers, we perform an estimation of their contribution in improving the overall video quality. This is in contrast with the JSVM–QL technique, which computes the quality contribution of each quality layer of each frame in the GOP in an exhaustive manner, thus resulting in an increased metadata computation time during the post-encoding phase.

Comparison with Transcoding: When compared to alternate bitrate adaptation mechanisms such as transcoding, scalable video coding incurs much less delay while adapting to varying channel conditions and client characteristics. Quantifying this reduction in end-to-end delay is difficult due to the fact that there is no standard method of transcoding. One possible solution is to implement a simple transcoder by cascading a decoder and an encoder. Here, each stream would be completely decoded and re-encoded according to the current available bandwidth. However, this would result in an unfair comparison since most commercial transcoders perform only a partial decoding and re-encoding in an effort to reduce the overall delay. Hence, such experiments have not been performed since optimizing transcoder performance is outside the scope of this work.

3.4.5 Snapshot of Algorithm's Performance

In this section, we demonstrate a snapshot of our algorithm's performance and compare it with JSVM–Basic and JSVM–QL. The Rush hour sequence extracted at 1800 kb/s is used as an example. Figure 21 shows the extraction of MGS quality layers from five consecutive GOPs (total of 40 frames) in this sequence. As Figure 21a

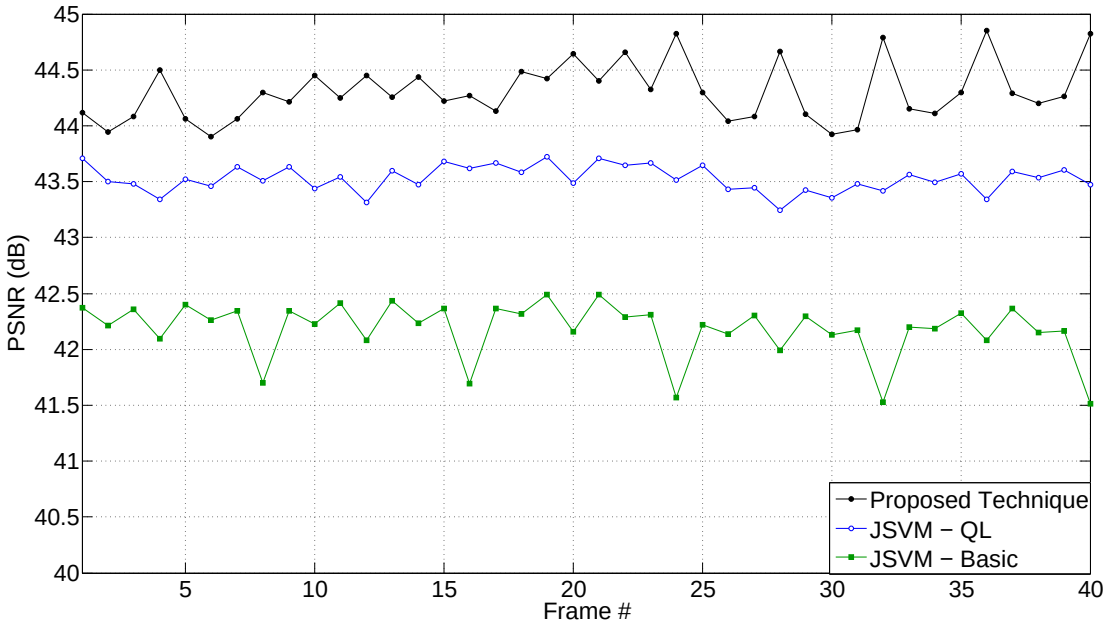
shows, the reconstructed video quality of every frame in all the five GOPs is better when extracted using our algorithm, compared to the extractions performed using JSVM–QL and JSVM–Basic. Figure 21b shows the size constraint and the size of each extracted GOP (in bytes) for all the three extraction techniques. The blue line denotes the size budget for each GOP. For an available bandwidth (B) of 1800 kb/s and a GOP size (N) of 8 frames and a frame rate (F) of 25 fps, the GOP size budget is calculate using Equation (3) as:

$$R_g = BN/F = 1800(8/25)(1000/8) = 72000 \text{ bytes} \tag{10}$$

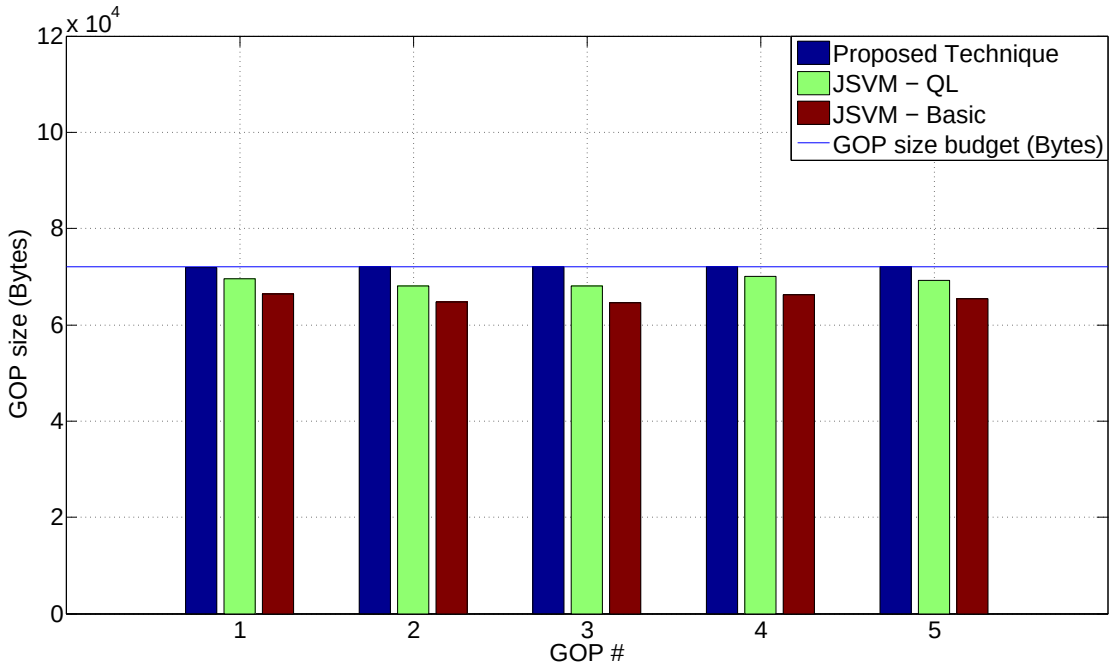
Any extraction algorithm should not exceed this size limit while extracting a GOP. As the figure shows, all the techniques lie below the limit for every GOP. Our technique uses more bytes than the other two techniques since it extracts more quality layers. However, the total size of each extracted GOP is still less than the allowed limit.

3.4.6 Estimated and Actual Distortions

In this section, we show the accuracy of the distortion values that are estimated by our algorithm during the decision making process of extracting MGS quality layers. The distortions are computed for the lowest and the highest quality layers for every frame predicted from lowest and highest quality reconstructions of their parents. Both these distortion values are proportionately scaled depending on the number of quality layers extracted for each of the parent. Finally, the total distortion reduction obtained by decoding the lowest and highest quality layers of a frame is distributed among all the in between MGS quality layers in proportion of their sizes. Hence, the distortion of each quality layer for every frame within a GOP is estimated and an RD optimal extraction of layers is performed that minimize this distortion. The details can be found in the flowchart in Figure 15 of the previous section. In Figure 22, we compare the estimated distortion with the actual distortion obtained on decoding the



(a) Quality (PSNR) of 40 reconstructed frames (5 GOPs) extracted using all the three techniques



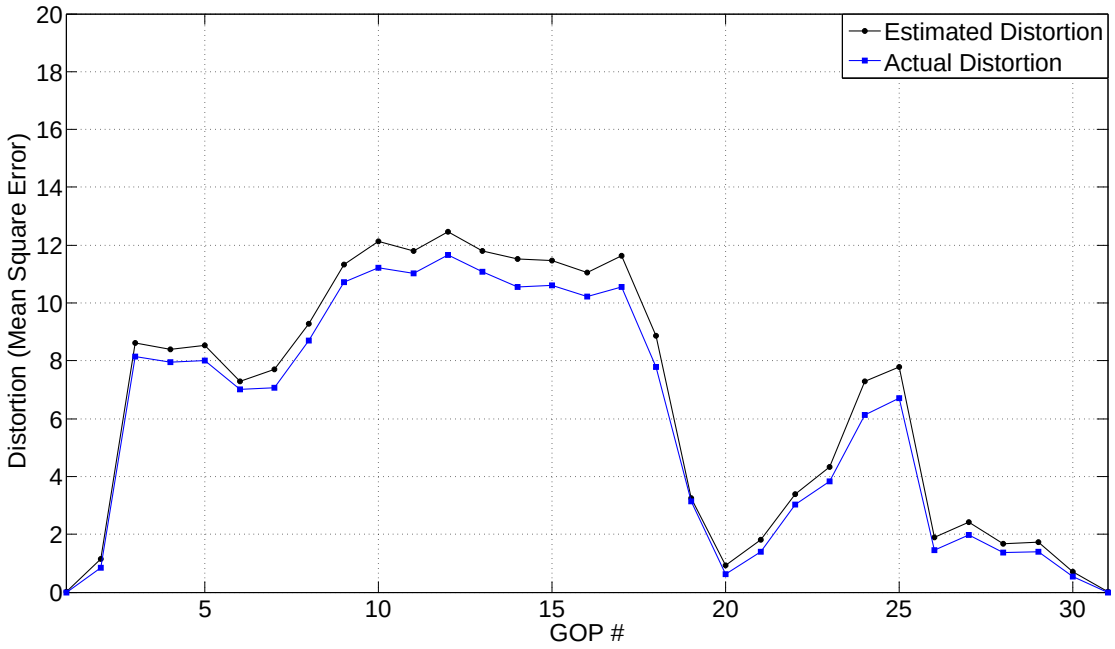
(b) Size of the GOPs extracted using all the three techniques

Figure 21: Snapshot of the performance of the proposed technique compared to JSVM – QL and JSVM – Basic for the Rush hour sequence extracted at 1800 kb/s.

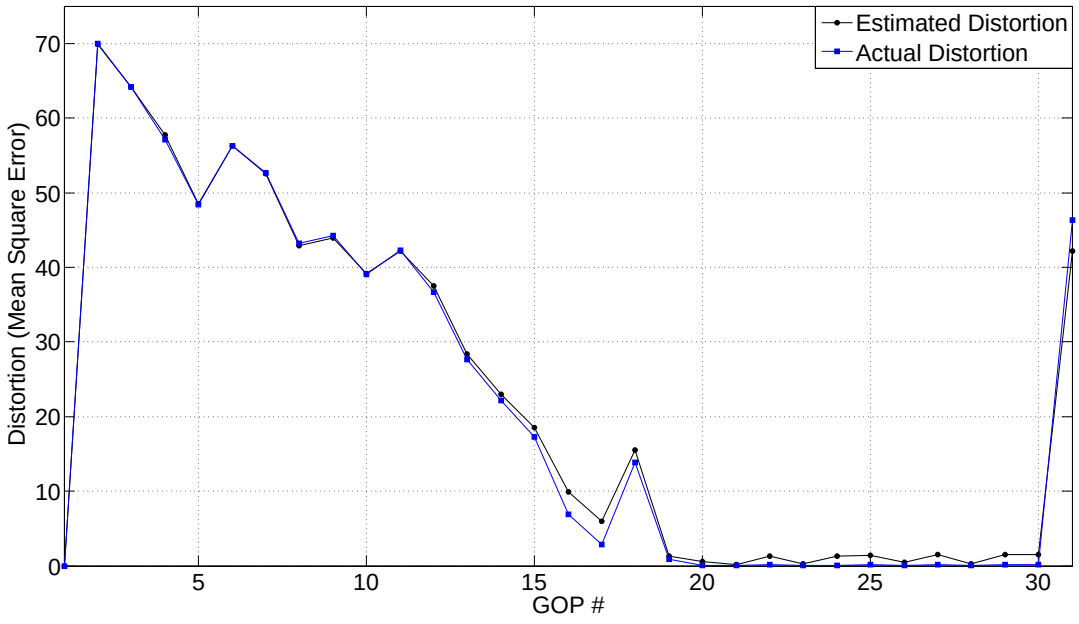
extracted sequences. The distortion measure used is mean square error. We show two examples: Mobile sequence from SET 2 (CIF) extracted at 2400 kb/s and Red kayak sequence from SET 1 (720p) extracted at 3000 kb/s. As we can observe from the graphs, our algorithm does estimate the mean square error quite accurately, given the number of limited decodings it performs and its estimations of the distortion reduction contributed by the in between MGS layers. We also see that the estimated mean square error follows the changes in actual distortion closely. This proves the rate-distortion effectiveness of our technique. Hence, the bitstreams extracted by our technique are indeed RD optimal.

3.4.7 Sample Frames

In this section, we show some sample frames to illustrate the difference in reconstructed video quality when extractions are performed using our proposed algorithm and JSVM–Basic. We show three examples. Figure 23 shows Frame # 1 of the Aspen sequence that has been extracted at 2000 kb/s. The regions with visible artifacts (mostly blockiness and blurriness) in the frame extracted using JSVM–Basic are shown in red circles. The veins in the leaf are blurred in JSVM–Basic extraction, whereas it is sharp in the frame extracted using our technique. Figure 24 shows Frame # 153 of the Red kayak sequence that has been extracted at 1750 kb/s. The smudge in the trees and blocking artifacts in the water are clearly visible in the frame extracted using JSVM–Basic when compared to the frame extracted using our technique. Figure 25 shows Frame # 385 of the Red kayak sequence that has also been extracted at 1750 kb/s. The blockiness appearance in the kayak and the blurriness in the waves are more prominent in the frame extracted using JSVM–Basic than in the frame extracted using our proposed algorithm.



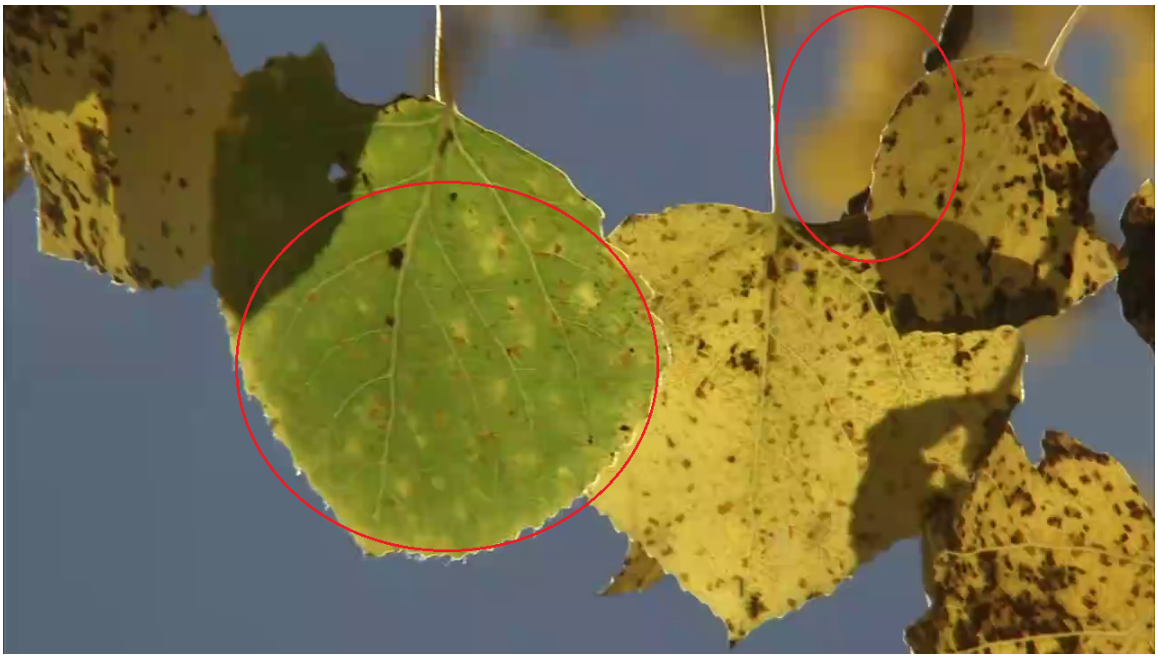
(a) Mobile sequence extracted at 2400 kb/s



(b) Red kayak sequence extracted at 3000 kb/s

Figure 22: Comparison of the estimated distortion and actual distortion in the extracted sequences.

(a) Extracted using the proposed technique



(b) Extracted using JSVM – Basic

Figure 23: Frame # 1 of the Aspen sequence extracted at 2000 kb/s.

(a) Extracted using the proposed technique

(b) Extracted using JSVM – Basic

Figure 24: Frame # 153 of the Red kayak sequence extracted at 1750 kb/s.

(a) Extracted using the proposed technique

(b) Extracted using JSVM – Basic

Figure 25: Frame # 385 of the Red kayak sequence extracted at 1750 kb/s.

3.5 Summary

In this chapter, we have focused on bitstream extraction techniques for scalable video coding based video streaming. First, we have discussed an SVC-based streaming application, namely the three-screen TV. An end-to-end system architecture using a home gateway for television broadcast and video-on-demand services has been described. Three-screen TV involves media delivery to a variety of client devices including HDTV displays, tablets, netbooks, smartphones, etc., that vary in their display sizes, processing power and network connectivities. The home gateway acts as an intermediate network node that receives a single scalable bit stream (SVC) from the service providers. SVC bitstreams consist of a number of scalable layers along the temporal, spatial and quality dimensions. The scalability is achieved by providing the ability to extract and decode partial portions of the stream corresponding to certain spatial, temporal and quality resolutions. The gateway extracts partial bitstreams according to each client's requirements and current available bandwidth conditions between itself and each of its clients.

With this application as the motivation, the problem of extraction of a scalable bitstream according to available channel resources has been formulated. The various dimensions (quality, temporal or spatial) along which extractions can be performed have been described and their pros and cons have been examined. The main challenge behind such an extraction process is how to extract a rate-distortion (RD) optimized stream at a certain spatial, temporal and quality resolution for a given available bandwidth in the channel. In other words, the extracted video stream must be the best possible quality video stream (in a rate-distortion sense measured through a metric such as PSNR) that can be obtained at that bitrate.

Current state-of-the-art extraction techniques like JSVM–Basic and JSVM–QL have been investigated. Next, our solution algorithm has been described to address the problem of RD-optimal bitstream extraction at a given bitrate. It consists of

three components: Computation of the quality contribution of each layer in the bitstream, signaling of that information and extraction based on this quality metadata at an intermediate network node. Our algorithm is based on estimating the distortion of each layer by performing a limited number of decodings ($2(1+\log_2 N)$) of the bitstream at the highest and lowest quality layers for each frame and predicting the quality contribution of the in between MGS layers according to their sizes. Priority IDs are assigned to base quality layers depending on their ease of concealment by their parents. The extraction process at the intermediate network node involves the evaluation of each candidate layer for extraction based on the estimated distortion reduction obtained by decoding that layer. The candidate that maximizes this reduction is selected. The estimates for current distortion of the extracted GOP is updated as new layers get extracted. The process continues till the available bandwidth is used up or all the layers have been extracted.

Experiments for MGS quality layer extraction, base quality layer extraction and metadata computation time have been performed on a number of sequences at a variety of spatial resolutions. Our results have been compared to existing techniques such as JSVM–Basic and JSVM–QL. On an average, our algorithm achieves a gain in video quality of about 1.5 dB over JSVM–Basic and a gain of about 0.5 dB over JSVM–QL. The maximum gain is about 4.0 dB when compared to JSVM–Basic and about 1.5 dB when compared to JSVM–QL. The time required for computing the metadata information during the post-encoding phase is 73% lesser for the proposed technique when compared with JSVM–QL. This huge reduction in metadata computation time along with the improvements in video quality make our technique a more preferred candidate than JSVM–QL and JSVM–Basic for use in real-time streaming applications. These results demonstrate the superiority of the proposed technique in delivering better video quality for a given bitrate while performing lesser number of computations for evaluating each layer's RD importance.

CHAPTER IV

SVC BITSTREAM EXTRACTION FOR CONFERENCING

In this chapter, we investigate scalable video coding based video conferencing applications and propose bitstream extraction techniques that maximize the reconstructed video quality under varying bandwidth conditions. Video conferencing belongs to a set of interactive applications that are constrained by tight end-to-end delay and jitter limits. Hence, these applications pose special challenges in the design of scalable communication systems. In the first section, an application of video conferencing in an enterprise environment is discussed. An end-to-end system architecture is proposed and an SVC based multipoint control unit (MCU) is described for bitstream extraction and coordination of traffic flow between the multiple parties involved in video communication. With this application as motivation, the problem of adapting the bitrate of real-time, SVC encoded conversational video to varying channel conditions is formulated with an aim of maximizing the decoded video quality. Solution approaches where extraction decisions are made over single and paired frames are proposed. This is followed by our main solution design and algorithm, which is based on paired-frame extraction using quality metadata information. The extraction decisions made by this technique are near RD optimal. Next, our technique is validated with experimental results. Paired-frame extraction using quality information shows a maximum quality increase of about 0.2 dB when compared with simple paired-frame extraction and an increase of about 1.3 dB when compared with frame-by-frame extraction. Finally, all our findings and results are summarized for the video conferencing application.

4.1 Application – Enterprise Video Conferencing

In this section, we look at the application of video conferencing over enterprise networks. We describe an end-to-end system architecture based on hybrid multipoint control units (MCUs). In such scenarios, we show how scalable video coding (SVC) is suited as the ideal solution for performing bitstream adaptations to changes in available bandwidth on the transmission channel. We also discuss alternate adaptation mechanisms and their disadvantages. We also validate the use of available bandwidth as a metric in such applications.

4.1.1 End-to-end System Architecture

Figure 26 shows the architecture of a typical interactive multimedia communication system in an enterprise [9, 46]. It depicts the communication between a corporate headquarters, two branches and an employee connecting remotely from home. Within each branch, participants connect to the session via a video conferencing room or their desktops, both having wired connectivity to the LAN. Some members also connect wirelessly through WiFi networks. All such intra-branch connections together form the core enterprise channel. The branches are interconnected using leased lines and VPNs through the Internet as shown in the figure. These inter-branch connections form the peripheral enterprise channel.

Each branch is equipped with one or more multipoint control units (MCUs) whose main function is to bridge the conference sessions involving members from that branch. The SVC-based MCU adapts the bitstreams that originate from its branch to changes in the available bandwidth in the peripheral enterprise channel. For employees connecting remotely from home, such an MCU is usually implemented in software. The MCUs from each participating branch coordinate session initiation, termination and control information for all the members of that branch. It acts as a central exchange point for all communication to and from that branch (similar to a

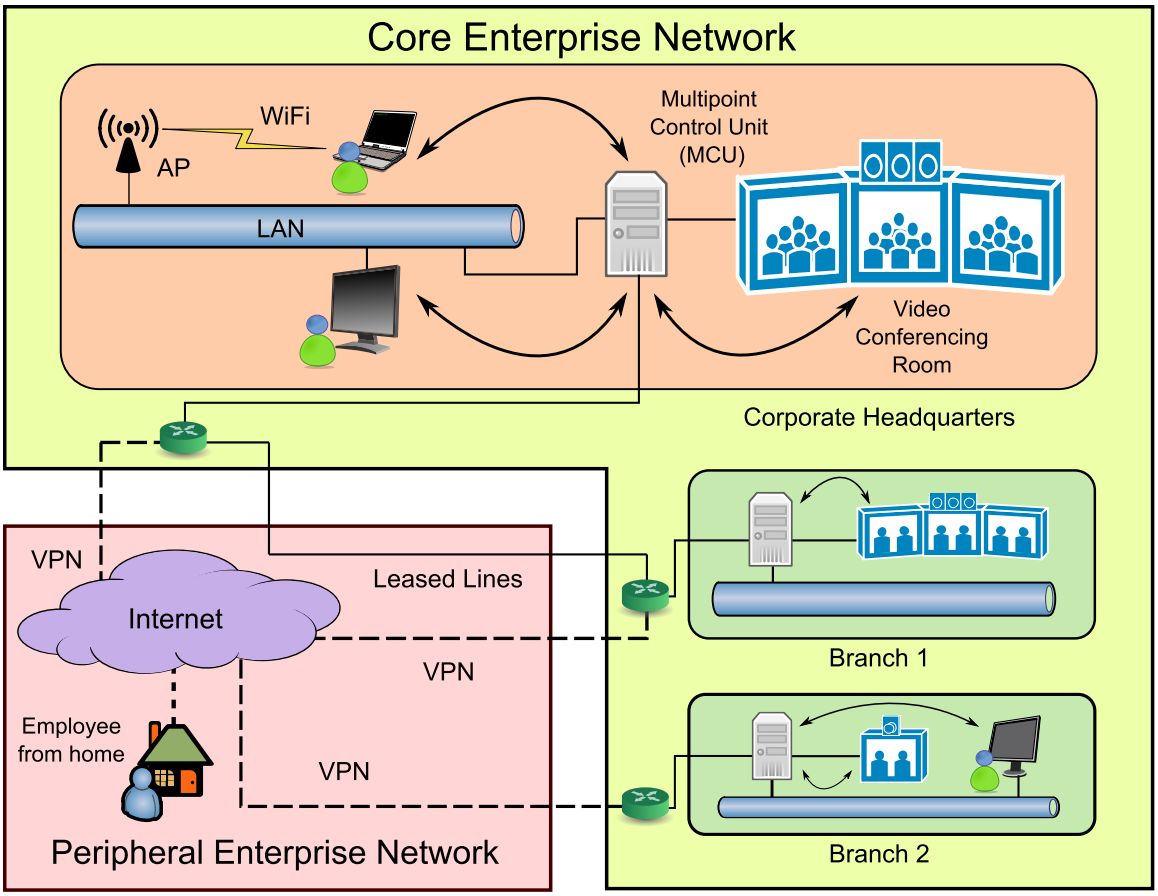


Figure 26: Architecture of an interactive multimedia communication system over enterprise networks.

home gateway in the one–way streaming scenario).

When the communication is purely intra-branch, a single MCU belonging to that branch manages the entire conference session. It distributes the streams received from each participant within the branch to all the other members located within the same branch. Since within the core enterprise network high QoS can be guaranteed for real-time traffic, the MCU prioritizes all the received media streams for real-time delivery. This eliminates the need to do bitrate adaptation within the core enterprise channel [71–74].

When the communication is inter-branch, a number of MCUs are involved in

coordinating the communication, one for each branch participating in the conference session. Inter-branch communications involve the peripheral enterprise channel (leased lines, VPNs through Internet, etc.), where the available bandwidth could be limited. Now the responsibilities of each MCU is multifold, which can be summarized as follows:

1. Distribution of the media data received from participants from other branches to the members on its core enterprise network.

2. Aggregation of the media data generated by participants within its core enterprise network and serving it to all the other participants across the peripheral enterprise channel through their respective MCUs.

3. Measurement or estimation of the available bandwidth of the channel between itself and each of the remaining MCUs.

4. Performing multiple adaptations of each outgoing video stream to suit the changes in available bandwidth between itself and each of the remaining MCUs.

5. Prioritization of the most important parts of outgoing video bitstreams. The degree of prioritization depends on the QoS levels on the peripheral enterprise channel.

6. Coordination of the conference session with respect to its core enterprise network (i.e., the branch to which the MCU belongs).

Provisioning and QoS are always limited along the peripheral enterprise channel due to the involvement of public Internet connections [71]. Hence, one of the main challenges faced by MCUs is maximizing video quality while transmitting on the peripheral enterprise channel. In Figure 26, this refers to all the inter-branch communications.

In our architecture, video conferencing sessions involving inter-branch communications require an MCU for each participating branch. This is a decentralized hybrid approach to bridging conference sessions that is in contrast to the common technique of managing an entire conference session among all the branches with a single MCU, whose location is chosen depending on the branch that initiates the session. The main problems with the centralized approach can be categorized into a number of key areas:

Timely Delivery: In the centralized approach, all the communication is relayed through a single MCU that is usually located in the branch that initiated the session. This is an important issue when the peripheral enterprise channels are involved, where the QoS guarantees are less effective. For e.g., in Figure 26, let us assume a centralized approach where MCU of the corporate headquarters is used for coordinating the conference session; then, all video traffic from Branch 1 must be sent to the headquarters from where it will be distributed to Branch 2. This will result in the video stream from Branch 1 traversing the peripheral enterprise network twice: first to reach the headquarters and then to reach Branch 2 from the headquarters. Given the limited effectiveness of QoS guarantees on the peripheral enterprise channel, these streams would be subjected to delays and packet losses. Hence, they might not satisfy the tight end-to-end delay constraints of interactive video and would result in poor performance. Distributing the MCU solves the problem by eliminating the need to relay all media traffic through a single MCU. Live video traffic from Branch 1 to Branch 2 is sent directly over the peripheral enterprise channel connecting the two, thus ensuring a much higher probability of timely delivery than the centralized approach.

Bitrate Adaptation: In the centralized approach, bitrate adaptation of streams is done at a single MCU based on the available bandwidth between itself and each of the participants. In the previous example, this would mean that bitrate adaptation

will be done for only one-half of the path (from MCU to Branch 2). For the path from Branch 1 to MCU, no adaptation is possible unless it is performed by the sender at Branch 1. Hence, if the bitstream adaptation process is moved to the sender (i.e., every participant in the video conferencing session), then bitrate adjustment can be done along every path connecting the participants. But doing so raises another issue. In our example of sending video data from Branch 1 to Branch 2 via the MCU in headquarters, the stream will undergo two sets of adaptation. The first one will be based on the peripheral enterprise channel between Branch 1 and the headquarters with the original encoded stream as input, and the second will be based on the channel between the headquarters and Branch 2 with the adapted stream as input. If the channel between Branch 1 and headquarters is poor but the channel between headquarters and Branch 2 is very good, the extra available bandwidth on the path between the headquarters and Branch 2 cannot be utilized since the stream received by headquarters would have been that of the lowest quality. Hence, the video quality received by a branch depends not only on the channel between itself and the headquarters but also on the channel between the sender of the video and the headquarters. This problem is readily solved by the decentralized approach, where the MCU in Branch 1 measures the available bandwidth between Branch 1 and 2 and directly sends its video stream to the MCU in Branch 2. Available bandwidth between Branch 1 and headquarters has no influence on the stream sent from Branch 1 to Branch 2 since it is not relayed through the headquarters anymore.

Communication Bottleneck: In the centralized approach, a single MCU coordinates the entire video conferencing session and also performs multiple bitrate adjustments for every stream. Hence, the MCU must be computationally powerful to handle heavy media traffic; otherwise, it would become the bottleneck in the communication system. Any failure in the MCU renders the entire communication inoperable. In our approach, the load is distributed among the various MCUs, where

each MCU needs to take care of the streams originating within its core enterprise network only. Distributing the MCUs avoids a single point of failure. Even if one of the MCUs fail, only that branch is disconnected and the remaining interaction can proceed without any hindrance.

The MCU decides on the layers that need to be extracted based on the available bandwidth conditions. It also performs the actual extraction process on the bitstream. In this sense, the MCU acts as the decision agent as well as the adaptation point [54] for the streams originating within its core enterprise channel. Another possible technique includes a participant-based subscription model, where MCU from each participating branch subscribes to a set of layers from each of the other MCUs. The number of layers subscribed depends on the average layer bitrates and the available bandwidth in the peripheral enterprise channel connecting the two MCUs [75]. Hence, for each video stream, the MCU on the receiver branch acts as the decision agent and MCU on the sender branch acts as the adaptation point. This method is not well suited for interactive video due to the following reasons:

1. Individual video frame data is known for its burstiness, and it might not adhere well to average layer bitrates.

2. Since the adaptation is performed at the sender's MCU after receiving the decision from the receiver's MCU, there will be an added latency in implementing the bitrate adjustment of the stream. Hence, the adaptation speed is reduced that affects system performance when available bandwidth changes at a rate faster than the adaptation speed.

4.1.2 Alternate Bitrate Adaptation Mechanisms

Other technologies suited for bitstream's rate adjustment include encoder-based rate control, multiple bitstream switching [49], transcoding [40, 50], and multiple description coding [52, 53]. Since the encoding is done in real time for video conferencing,

rate control of a single non-scalable stream can be done at the encoder by adjusting the encoding parameters like quantization step size, picture type, etc. However, such a control is possible if the communication involves only two members. When there are multiple participants in a video conference, there is no one way to satisfy the bitrate requirements of all the clients simultaneously except encoding a stream for each client with bitrate adaptation based on the feedback from that client. When the number of participants are more than three or four, this is highly impractical because of the delays involved in encoding multiple streams. A practical option is to encode a fixed number of bitstreams at different bitrates and switch among them depending on the channel conditions. Switching can occur only at designated points in the stream, such as I-pictures. However, I-pictures are used sparingly as their frequent use reduces compression efficiency. This results in delayed switching that reduces the reaction speed to changes in bandwidth. The granularity achieved is coarse depending on the number of streams the encoder can encode in real time (usually two or three). On the other hand, SVC requires the encoding of a single stream that is slightly more complex than that of a non-scalable stream. Bitrate adaptation to changes in available bandwidth of the peripheral enterprise channel is handled separately at the MCU.

Transcoding at the MCU is another choice, but it is a computationally expensive and a high-delay operation. Hence, it is not a viable option for interactive video communication with tight end-to-end delay constraints. In multiple description coding (MDC) approach, the content is encoded into multiple descriptors, each of which is independently decodable. When more than one descriptor is received, the quality is enhanced. MDC's success depends totally on path diversity, which can be realized in the peripheral enterprise channel by forming an application-specific overlay network of all the branches over the Internet. Each intermediate branch acts as a relay node for the video data, thus forming multiple paths between each sender and receiver.

However, as we have seen before, relaying video data through multiple peripheral enterprise channels is detrimental to its timely delivery. Also, the redundant information among the descriptors reduces the compression efficiency. These factors make MDC unsuitable for interactive communication. Since some form of QoS support is already provided by the enterprise, SVC forms the ideal candidate in such circumstances. The base layer can be guaranteed timely delivery by bandwidth provisioning, and the enhancement layers can be transmitted depending on the available bandwidth. The performance of such SVC based systems rely on the timely delivery of at least the base layer. Other forms of error control like retransmissions and forward error correction (FEC) [58] incur additional delays that make them unsuitable for interactive communication.

4.1.3 Available Bandwidth Metric

For interactive media traffic, network state description in terms of available bandwidth [76,77] is the most suitable metric since it captures the idea of the source video bitrate needed to reach the destination on time for decoding. The sender's MCU can employ a number of ways to measure/estimate the available bandwidth information. Since the communication is in both directions, the MCU can observe the packet arrival rate from the other branches and then predict the available bandwidth for the reverse path. This assumes uplink and downlink bandwidth symmetricity, which is usually true for enterprise networks. Otherwise, each MCU can explicitly measure the uplink and downlink available bandwidth on each of its peripheral enterprise channel as shown in [75]. It can also estimate the available bandwidth by modeling it as a Gaussian process as shown in [78].

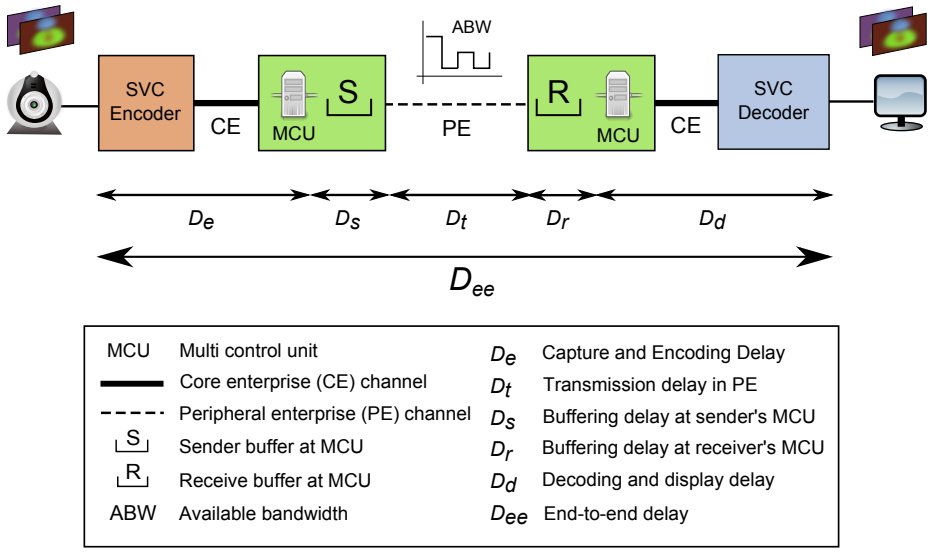


Figure 27: Delay components of a real-time video communication system.

4.2 SVC Bitstream Extraction – Preliminaries

In this section, we formulate the problem of rate-distortion optimal SVC bitstream extraction for real-time interactive applications, which are characterized by tight end-to-end delay and jitter constraints. First, we perform an end-to-end delay analysis of such an interactive video conferencing system and study its various delay components. Then, we examine their differences when compared to one way streaming applications that were studied in the previous chapter. We identify the new challenges that arise in the interactive scenario and reformulate the bitstream extraction problem and develop solutions for performing a rate-distortion optimal extraction suited to such conversational applications.

4.2.1 End-to-end Delay Analysis of a Conferencing System

Based on the architecture described in the previous section, Figure 27 shows the various delay components of a real-time video communication system. The total end-to-end delay (D_{ee}) incurred by a frame is a "mouth to ear" delay, i.e., from capture

to display. It is the sum of all the delay components in the system:

$$D_{ee} = D_e + D_s + D_t + D_r + D_d \tag{11}$$

Acceptable values of D_{ee} are in the range of 150 ms to 350 ms. Values above 400 ms are unacceptable [15] since maintaining interactivity becomes a problem. Let us analyze these delay components in the context of real-time interactive communication. Let the maximum frame rate of the sequence be F frames per second (fps). Then, the sampling interval or the duration for which each frame is displayed, indicated by δ , is given by:

$$\delta = 1/F \tag{12}$$

The total end-to-end delay (D_{ee}) can be split into two broad categories: fixed delay (D_{fix}) and variable delay (D_{var}). Delay due to capture, encoding, decoding and display can be considered as fixed delays. The encoding and decoding delays for each frame depends on the frame's complexity and the encoding modes used. Interactive video sequences have very similar encoding complexities due to minimal motion between frames, very rare scene changes, and use of similar encoding modes and prediction structures for all video frames. Hence, the encoding and decoding delays can be categorized as fixed-delay components. Delay due to queueing at the sender and receiver buffers and transmission in the peripheral enterprise channel are variable components.

The transmission delay (D_t) in the peripheral enterprise channel is dependent on the available bandwidth in the channel. During low bandwidth conditions, it leads to additional buffering delay (D_s) for successive frames at the sender's MCU. The receiver buffer queues the received frames for a duration of D_r before decoding. Jitter compensation and frame reordering occurs here. The receiver buffering delay depends on the frame's arrival time, which in turn depends on the available bandwidth conditions. The higher the bandwidth of the peripheral enterprise channel, the sooner

the frame arrives and hence, spends more time in the receiver buffer. During lower bandwidth conditions, the frames arrive later and spend lesser time in the buffer before it is decoded.

Let the size of the receiver buffer be represented as S_r. For a frame with a planned transmission delay of δ (corresponding to a jitter-free transmission with video bitrate = channel bandwidth), the buffering delay is $S_r - \delta$. The decoder waits for this duration before picking up the first frame of data for decoding and display. Hence, this delay is the start up delay for the conferencing application. Such a delay is usually limited to one or two frame durations due to the tight end-to-end delay and jitter constraints. Once the decoder has decoded the first frame, it continues decoding at a constant rate by picking up one frame every δ seconds from the receiver buffer. Decoding at a fixed rate (in terms of number of frames decoded per second) ensures constant end-to-end system delay and eliminates motion rendition issues. Such a decoder is said to operate in constant-delay mode [79]. This is in contrast to operating the decoder in low-delay mode where each frame is decoded immediately on reception, i.e., no buffering is involved. This results in end-to-end delay variations between successive frames that cause frequent "jumps" and "drags" in motion rendition affecting the temporal smoothness of the video sequence.

Choosing the correct value of S_r is complicated. Setting a high size for the receiver buffer (S_r) has the advantage of compensating large variations in frame arrival times but it increases the total end-to-end delay of all the frames. Since interactive video frames have tight upper bounds on maximum tolerable end-to-end delay, this leads to performance degradation. Hence, the size of the receiver buffer is set equal to the longest tolerable transmission delay (D_{tm}). When a frame suffers a transmission delay greater than D_{tm}, it is considered to be too late for decoding and discarded. If a frame incurs a transmission delay (D_t) less than D_{tm}, then it is queued in the

receiver buffer for a duration D_r such that:

$$D_r = D_{tm} - D_t \tag{13}$$

A frame that suffers the maximum tolerable transmission delay of D_{tm} will be picked up for decoding immediately upon arrival. In other words, its buffering delay (D_r) is zero. Hence, decoder implementations have S_r set to two or three frame intervals usually. This achieves a good compromise between jitter compensation and increase in end-to-end delay.

4.2.2 Video Conferencing and Streaming: A Comparison

There are a number of differences between video conferencing and streaming applications in terms of sequence content, encoding parameters, end-to-end delay, jitter requirements, etc. Interactive sequences differ from professionally-encoded sequences like movies, sport events, etc., in the type of content being captured. Most interactive sequences are characterized by very little motion between frames and rare scene changes. Also, the camera focus would always be on a single or a group of people. It involves high spatial details, especially when presentations are being delivered. Real-time encoding of such sequences is a challenge. Encoding complexity and delay are minimized by avoiding B-frames and limiting the motion search range. Inter prediction is done only using P-frames, and hence the encoding order is same as the display order. For a GOP size of eight frames at four temporal layers ($T = 0, 1, 2, 3$), Figure 28 shows a zero-delay encoding structure commonly employed (used in all our video conferencing experiments) [46]. The zero-delay encoding structure allows immediate encoding of a frame once it is captured. The predictions used for encoding the frame come from pictures that have already been encoded. The delay structure (hierarchical prediction B-pictures) used for one-way streaming, as shown in Figure 9, has a structural delay of $N\delta$, where N is the GOP size in frames and δ is the frame interval duration. Such a high-delay structure cannot be used for real-time encoding

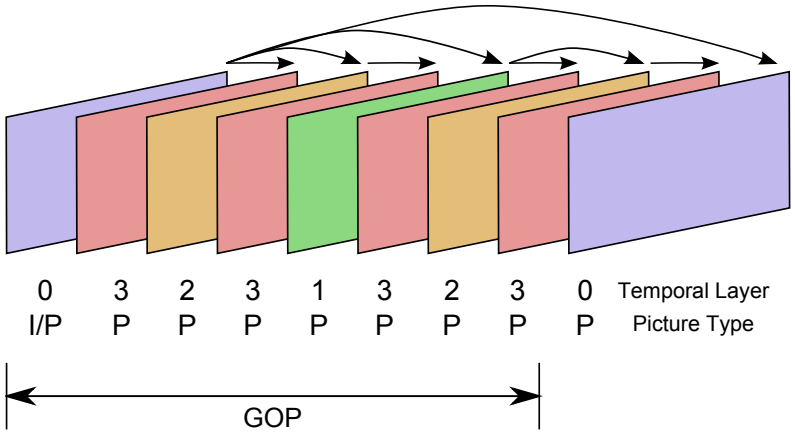


Figure 28: Zero-delay encoding structure with a GOP size of eight frames at four temporal layers.

of conversational video sequences as the interactive applications are constrained by a tight upper bound on end-to-end delay. Due to low motion between successive frames, temporal predictors are a better choice than inter-layer predictors for encoding the enhancement layers [5]. Frequent intra refresh (IDR pictures) is not required since there are no scene changes. The key aspect in two–way applications is the fact that there is no GOP based processing due to the tight end-to-end delay constraints. Each frame is treated either individually or in pairs.

4.2.3 Bitstream Extraction: Problem Formulation

Every δ, video frames are captured, encoded and transmitted one by one from the participant to its MCU. These frames arrive every δ at the MCU, where the SVC bitstream extraction process is carried out. The MCU measures the available bandwidth in the channel between itself and the remaining participants (or their controller MCUs). Based on the current channel conditions, the MCU performs an RD optimal extraction of the SVC bitstream. The main challenge in applying the RD optimal extraction process developed for the streaming scenario in the previous chapter is the fact that the MCU does not have enough frames of data (usually the decision making window is 1 GOP of frames in one–way streaming) to select MGS quality

layers belonging to several temporal layers within a GOP. Since frames arrive at the MCU only every δ, the MCU needs to wait for $N\delta$ s to accumulate one GOP of data so that it can apply an RD optimal extraction process for the various quality layers in the entire GOP. But the delay incurred in this extraction process is unacceptable for interactive communications. Hence, the MCU needs to reduce the decision making window to that many frames which can be delivered within the end-to-end delay limits.

Let us formulate the problem mathematically for a single spatial resolution. Let N be the GOP size in frames, Q_m be the maximum encoded quality layer, F fps be the frame rate, and B bits/s represent the currently available bandwidth in the channel. The frame interval is represented as $\delta = 1/F$. If the maximum tolerable transmission delay (for jitter compensation) is represented as D_{tm} and the average transmission delay is represented as D_{tg}, then the number of frames in the decision window of the MCU is can be represented by n, such that:

$$n = D_{tm}/D_{tg} \tag{14}$$

The average transmission delay (D_{tg}) corresponds to a jitter-free transmission and hence, it is set to the frame interval δ so that the bitrate of the delivered stream corresponds to the bitrate of the channel. Since input frames arrive at the MCU from the sender at a frame rate of δ, it is necessary that the transmission delay of each frame be limited to δ to ensure that the transmission is jitter free. When the actual transmission delay exceeds D_{tg}, this appears as jitter and has to be compensated at the receiver buffer. When D_{tm} is set to D_{tg}, n becomes one, i.e., when the maximum tolerable transmission delay is set to the mean transmission delay of δ, the decision window is a single frame. The advantage in a single frame based decision is the fact that it ensures jitter-free transmission and does not require the receiver to implement a jitter-compensation buffer. Absence of dejitter buffering reduces the overall end-to-end delay for every frame. Hence, it increases the interactivity in the

video conferencing application. The main disadvantage is that decisions made over a window of single frame are not rate-distortion optimal and the reconstructed video quality tends to be poor.

When a jitter compensation of δ is provided at the receiver, the tolerable maximum transmission delay can be set to twice the mean transmission delay, and hence n becomes two. Now, the MCU has a window of two adjacent frames at different temporal layers over which it can make its extraction decisions. This leads to a more informed and a better rate-distortion optimal extraction when compared to decisions made from single frames. However, this also increases the end-to-end delay of the frames by δ but this is not a problem as long as we are within the allowable limits of interactive communication. Increasing the value of n further is not advisable since it will cause the jitter compensation buffer to grow in size that will result in higher end-to-end delays and exceed the acceptable limits of interactive communication.

The bit budget (R_n) for a set of n frames at a frame rate of F fps and an available bandwidth of B bits/s can be computed as:

$$R_n = nB/F \tag{15}$$

The number of quality layers in a frame is $Q_m + 1$ since quality layers begin from zero. Hence, the total number of quality and temporal layers in the set of n frames (assuming there is one spatial layer) is calculated as:

$$L_m = (Q_m + 1)n \tag{16}$$

If each of these layers were given an absolute layer ID (L), then the range for L is $0, 1, 2, \ldots L_m - 1$. The value of temporal ID (T) is in the range $0, 1, 2, \ldots \log_2 n$. The value of Q is in the range 0,1,2,$\ldots Q_m$. Let Size() represent the function that computes the size in bits of its input argument. Let S be a layer in the set of n frames. It is represented by its temporal ID t, quality ID q and absolute layer ID l,

i.e.,

$$S \equiv \{t, q, l\} \tag{17}$$

Let Δ represent the set of all layers S that when assembled together form a partial bitstream that represents the n frames of data and conforms to the SVC standard. Let $\Gamma(R_n)$ denote the set of all possible Δ that are of size less than or equal to the allocated bit budget (R_n) for n frames, i.e.,

$$\Gamma(R_n) \equiv \left\{ \Delta | \sum_{S \in \Delta} \text{Size}(S) \leq R_n \right\} \tag{18}$$

We are interested in the member Δ_{opt} belonging to $\Gamma(R_n)$ that minimizes the distortion function Dist(). This function computes the distortion (e.g. MSE) after decoding the stream represented by its input argument with respect to the source stream. It uses the available maximum quality reconstruction as the source stream while computing the distortions.

$$\Delta_{opt} = \underset{\Delta \in \Gamma(R_n)}{\text{argmin}} \ \text{Dist}(\Delta) \tag{19}$$

Equation (19) represents the problem of RD optimal SVC bitstream extraction for video conferencing applications, which we solve in the following sections.

4.3 SVC Bitstream Extraction – Solutions

This section proposes solutions to the problem of optimal bitstream extraction at a given bitrate for video conferencing applications. We examine extraction techniques that use single frame and frame pairs in their decision making windows. The pros and cons of both these techniques is presented and this is followed by our proposed technique, which is based on paired-frame extraction using quality metadata information. The structure of the algorithm is similar to the one proposed for one-way streaming but has some key differences.

4.3.1 Frame-by-frame Extraction

The MCU receives coded frames from the sender at the rate of one frame every δ. One frame of data is composed of the base quality layer and higher quality layers at various spatial layers for that frame. In the frame-by-frame extraction technique, the MCU extracts the frame as soon as it is received. Since the decision window is only one frame, the order of extraction is simply based on the quality layer IDs. As the SVC standard mandates that all lower quality layers be extracted before extracting a higher quality layer, the extractor starts with the base quality layer and extracts the MGS quality layers one by one in the order of increasing Q. This process is illustrated with a flowchart in Figure 29 for a single spatial layer resolution. As shown in the figure, the extractor is invoked as soon as each frame is received at the MCU. It takes three arguments, namely the size of the frame (with all its quality layers), temporal layer ID of the frame (needed to verify its dependency on past frames), and the available bit budget to extract the frame. The function checks the available bit budget (R_f) and performs the extraction for the frame in increasing order of the quality IDs after verifying the fact that their parents have already been extracted so that the SVC bitstream is still conforming to the standard.

The advantages of this technique include straight forward implementation and a jitter-free transmission. Since every extracted frame of data is assured to be transmitted within the frame interval of δ, there is no variation in arrival time of the packets at the receiver (jitter) and hence, the receiver does not need to implement jitter compensation buffer. The absence of queueing delay reduces the overall end-to-end delay incurred by the video frame. However, the biggest disadvantage of this technique is that the extraction is not RD optimal and the reconstructed video quality is not maximized. Another shortcoming of this technique is that the extraction happens independent of temporal importance as decisions are made on a frame-by-frame basis. As a result, if the base quality layer of a frame with temporal ID 1

(i.e., $T = 1, Q = 0$) is too big to fit within the allocated bandwidth for the interval of δ, then no layers are extracted for this frame, i.e., this frame is skipped. When future frames at higher temporal layers in the GOP arrive, they cannot be extracted too since dependency conditions would not have been satisfied as their parent frame at $T = 1$ has not been extracted previously. Hence, this technique of frame-by-frame extraction is most suited for high available bandwidth conditions, for e.g., core enterprise networks where QoS techniques are implemented to assure sufficient bandwidth to transmit atleast the base quality layer of each frame within the interval of δ. Figure 30 shows the typical order of layer extraction using this technique for a GOP size of eight frames with four temporal layers, five quality layers and one spatial layer. It can be noticed that the transmission delay (D_t) of each extracted frame is limited by δ, thus ensuring a jitter-free transmission.

4.3.2 Paired-frame Extraction

In the previous frame-by-frame extraction, the quality of the reconstructed video frame is not optimal since each frame is extracted independently with no attention to relative temporal importance (parent-child relationships among frames). This can be overcome by increasing the decision making window to include more frames from different temporal layers. In paired-frame extraction, the decision window uses two adjacent frames in display order. The first frame is usually the parent frame and the second frame is the child frame. From Figure 28, the first I/P picture from temporal layer 0 and the next P picture from temporal layer 3 form the first pair. The next pair consists of pictures from temporal layers 2 and 3. This way it can be ensured that the second frame in the pair is always at the highest temporal layer of the sequence and hence, is never used as a prediction parent of any other frame in the sequence.

Each of the frames used in the extraction's joint decision process arrive at the extractor every δ. Hence, the extractor needs to wait for δ after the first frame in

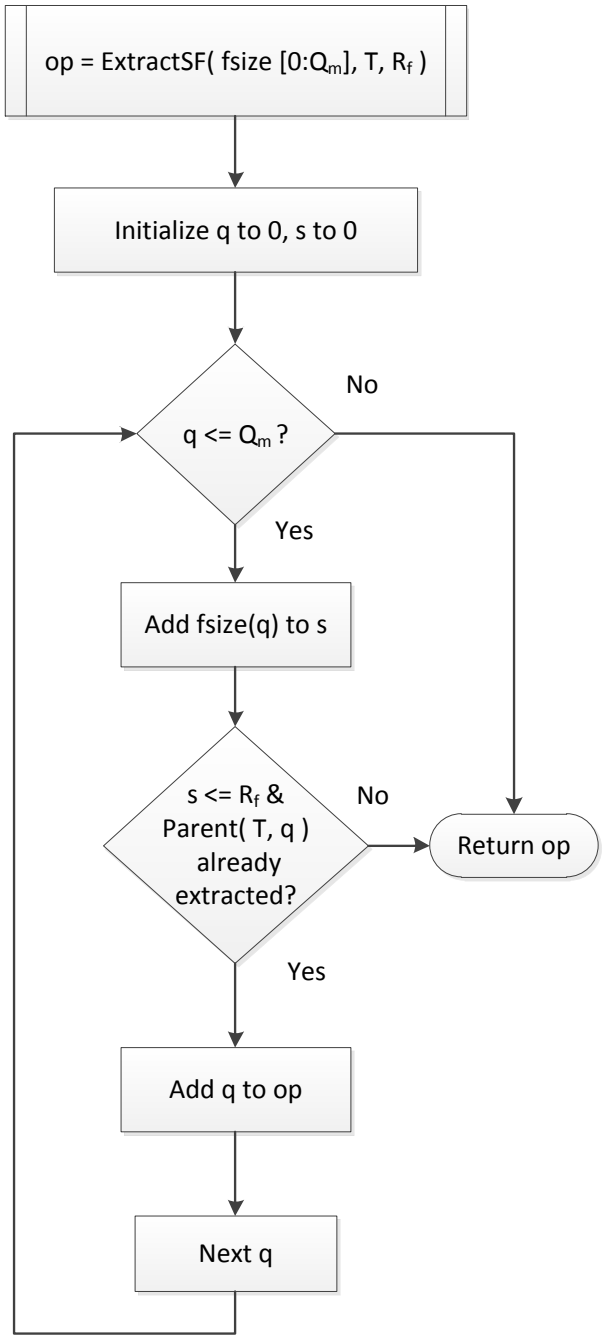


Figure 29: Flowchart for frame-by-frame extraction.

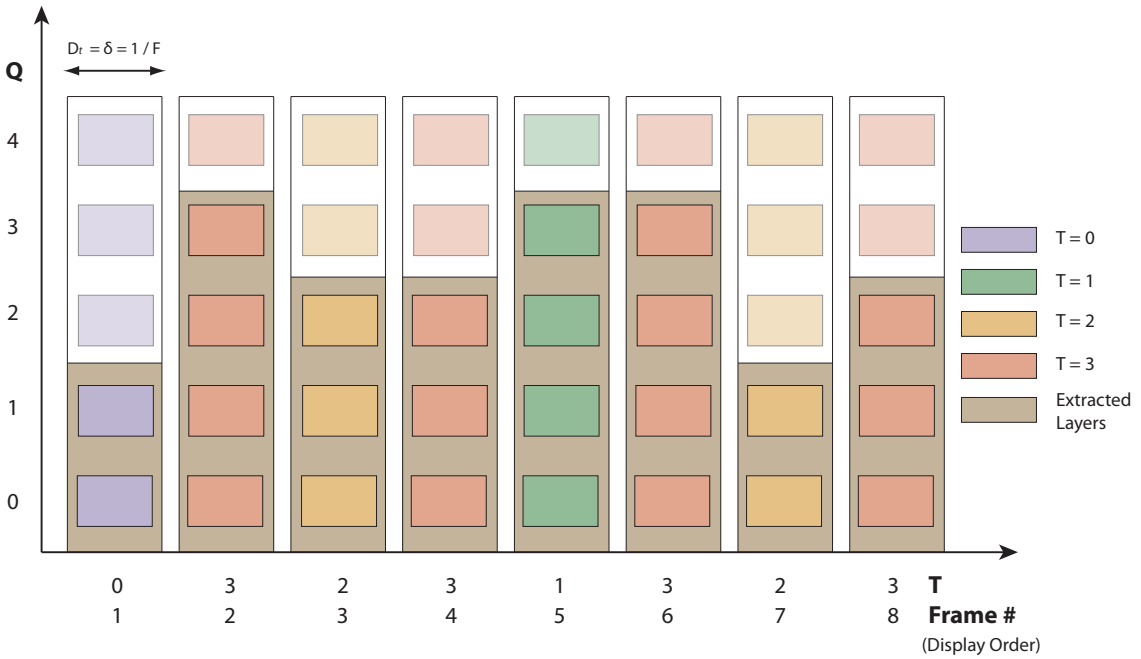


Figure 30: Typical order of layer extraction using frame-by-frame extraction for a GOP size of 8 ($T = 0, 1, 2, 3;\ Q = 0, 1, 2, 3, 4;\ D = 0$).

the pair is received so that it can accumulate both frames and then decide the layers to be extracted. This will add δ to the total end-to-end delay of each frame which may not be acceptable. Hence, the extractor starts extracting the first frame of the pair as soon as it is received. This is done in a manner similar to frame-by-frame extraction. But the difference is that paired-frame extraction continues the extraction of the current quality layer of the first frame even when the bandwidth limit for a single frame is reached. For e.g., let the bit budget for sending one frame be R_f and the size of all quality layers of the current frame up to the layer q be less than R_f and the size of the current frame up to quality layer $q + 1$ be greater than R_f and less than $2R_f$. Then, frame-by-frame extraction will stop the extraction at layer q but paired-frame extraction will include layer $q + 1$. Of course, the additional bits used come from the allotted bit budget for the second frame in the pair. This reduces the allotted bit budget for the second frame. By limiting the bit budget to $2R_f$ for the first frame in the pair, we ensure that the additional bits used do not go beyond

what is allocated for the pair. Under very low bandwidth conditions, it may happen that the first frame uses the entire $2R_f$ and hence, there is no bits left for the second frame in the pair, in which case it is dropped.

The transmission delay incurred by the first frame is represented as D_{t1} and is given by $\delta + \beta$ where β is the extra transmission delay incurred due to the sending of the additional quality layer. If the size of the extracted frame is S_e and the bitrate of the channel is B bits/s, then

$$\beta = (S_e - R_f)/B \tag{20}$$

The maximum transmission delay that can be incurred by the second frame in the pair is also reduced by the same amount of β so that its transmission delay(D_{t2}) must be less than or equal to $\delta - \beta$. β lies in between 0 and δ, both inclusive. This limits the maximum transmission delay (T_{dm}) to 2δ. Hence, the maximum possible variation in arrival times (i.e., jitter) at the receiver is also δ. At the receiver, this requires the presence of a receive buffer of size 2δ so that jitter compensation can be performed for all delay variations up to δ.

The advantage of paired-frame extraction is that the reconstructed video quality will be better than that of frame-by-frame extraction. With the pairing of frames, temporal importance of a frame forms a part of the decision process which was not the case with frame-by-frame extraction where all frames were treated equally. The structure of the pair is such that the frame at a lower temporal layer is always the first frame in the pair and hence, when it uses more bits than allocated R_f (these extra bits come from the allocation for the next frame in the pair), it would result in improvement of its quality as well as the quality of its child (the next frame in the pair and other frames in future pairs). This becomes especially significant when the base quality layer of the first frame requires more bits than the allotted R_f. Paired-frame extraction would result in extraction of the base quality layer of the temporally more important frame (first frame in the pair) at the expense of using extra bits originally

allocated for the second frame in the pair. Once the base quality layer of the parent frame in the pair is extracted, further extraction of the child frame in this pair can proceed as long as there is left over bit budget. This is not possible in frame-by-frame extraction where the base quality layer of the first frame would not have been extracted since its size was greater than the allotted R_f and hence, none of the layers of the second frame or future child frames in the GOP can be extracted even if the bit budget is available to extract them.

The disadvantage of this technique is the possibility of introduction of jitter in the range of 0 to δ. This requires a jitter compensation buffer at the receiver that is able to compensate jitter upto δ. This is not a problem since almost all decoders have such buffers usually of size δ, 2δ, or more. With such a buffer in place, the end-to-end delay also goes up by an amount δ. These increases are acceptable as long as they are within the range of interactive communication limits [15].

The working of the paired-frame extraction algorithm is illustrated in the flowchart in Figure 31. The extractor is invoked as soon as each frame is received by the MCU. It takes three arguments as its input, namely the size of the frame with all its quality layers, the available bit budget for the frame, and the frame's temporal layer ID. The extractor checks the temporal ID. If it is equal to T_m, it concludes it is the second frame in the pair. Otherwise, it is the first frame in the pair. The allocated bits (R_f) depends on the frame's position in the pair. For the first frame ($T < T_m$), $2R_f$ is allocated and for the second frame ($T = T_m$), a budget of R_f is allocated. Then extraction of the frame is performed starting from the lowest quality layer. At every step, it checks whether the allocated bit budget has been reached. If it is the pair's first frame, it allows the size of the last quality layer extracted to exceed R_f. The extra bits used (`r`) is updated so that when the extractor is again invoked upon the reception of the second frame in the pair, the value of R_f passed to the function has already been reduced by the extra bits (`r`) that has been used up during the extraction

of the previous frame in the pair. Hence, while extracting the second frame in the pair, the bit budget is limited to the updated value of R_f. Figure 32 shows the typical order of layer extraction using this technique for a GOP size of eight frames with four temporal layers, five quality layers and one spatial layer. Notice the transmission delay (D_t) of the first frame in the pair is $\delta + \beta$ and correspondingly for the second frame it is $\delta - \beta$.

4.3.3 Paired-frame Extraction using Quality Information

In the previous section, we saw that paired-frame extraction can give better reconstructed video quality than frame-by-frame extraction because of its decision window consisting of a pair of adjacent frames and the preference it gives to temporally more important frame in the pair by allowing it to exceed its allocated bit budget and consume extra bits from the second frame in the pair, which is at a higher and lesser important temporal layer. However, the decision to consume extra bits over the allocated bit budget is done for every first frame in every pair. This decision may not be the most RD optimal decision since it is done in a content-independent fashion. Hence, we make the paired-frame extraction technique better by making this decision in a more content-dependent way, i.e., by using quality metadata information similar to the one developed for the one-way streaming extraction.

The block diagram in Figure 33 illustrates an end-to-end video conferencing system based on SVC. The system is symmetric with respect to each of the participants in the conferencing session. From each user's perspective, the key blocks in the system include the process of encoding, post-encoding, extraction and decoding of bitstreams. An SVC encoder takes an uncompressed YUV stream as its input and generates an SVC bitstream with a certain number of quality, spatial and temporal layers that has been set while configuring the encoder. This is followed by a post-encoding process where the stream is extracted with various layers and decoded in order to evaluate

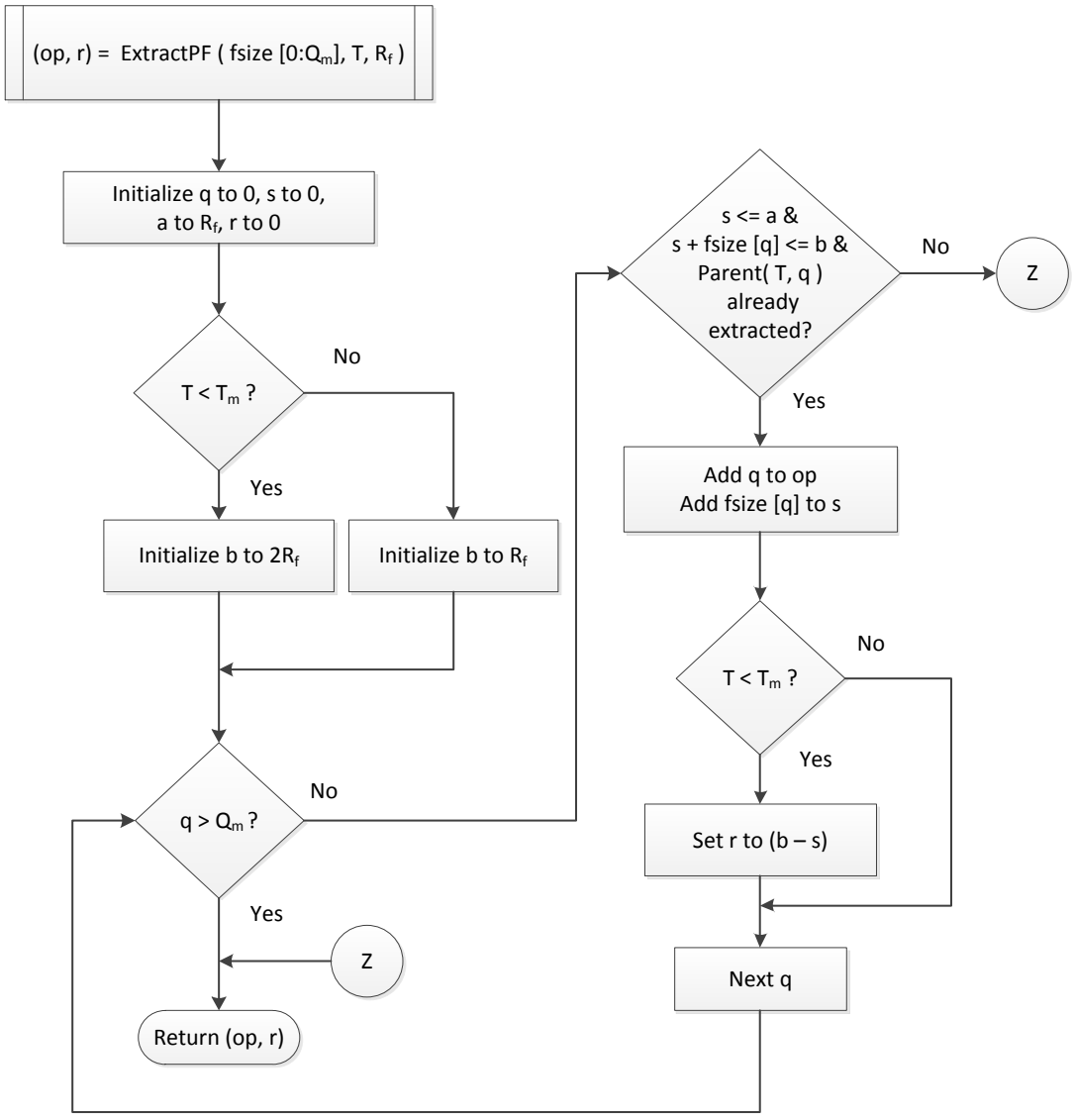


Figure 31: Flowchart for paired-frame extraction.

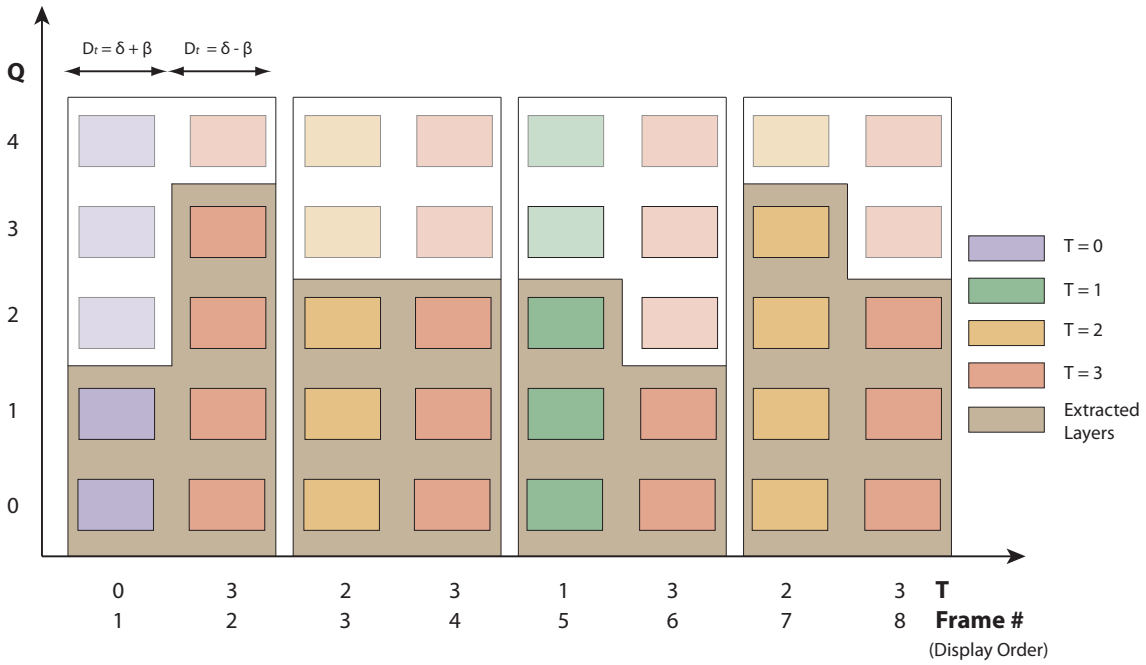


Figure 32: Typical order of layer extraction using paired-frame extraction for a GOP size of 8 ($TID = 0 \ldots 3, QID = 0 \ldots 4, DID = 0$).

the quality contribution of each layer. Since decoding of bitstreams is involved in evaluating each layer's importance, carrying out this process at intermediate nodes along the network path (like the MCU) will result in enormous end-to-end delays and hence, make the extraction process unsuitable for interactive applications. Hence, the layer quality contributions must be computed as a stand alone process. Usually, it is handled as a post-encoding process for video conferencing applications. Once computed, this quality information is stored in the NAL unit header of the bitstream or as SEI messages [45]. This relieves the extractor located at an intermediate network node (such as the MCU) from decoding the bitstreams and computing quality contributions. By simply looking at the NAL unit header and the SEI messages, the extractor can identify each layer's importance and extract according to the available bandwidth in the channel. Once extracted, the video streams are transmitted on the network and are finally decoded by the other participants in the video conferencing session.

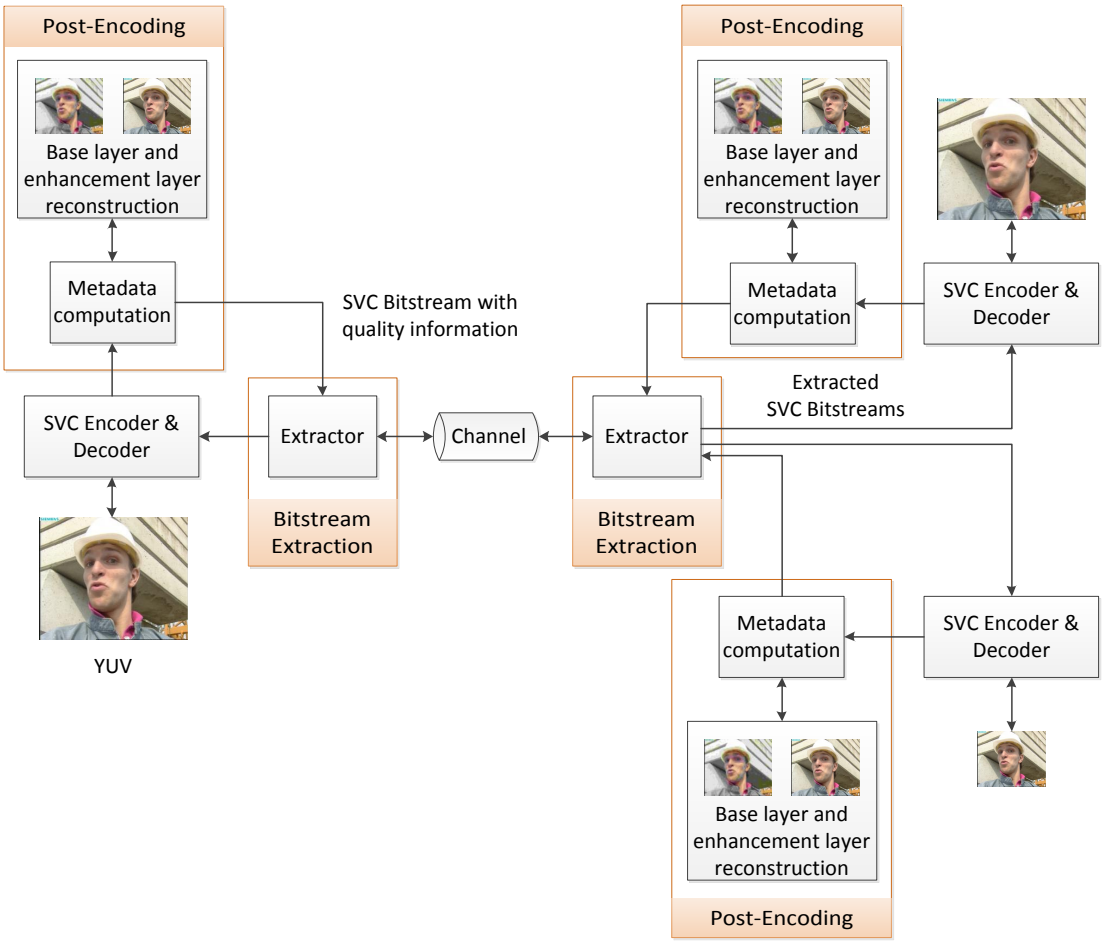


Figure 33: SVC-based conferencing system – End-to-end block diagram.

The blocks that are improved by our algorithm are those of post-encoding and extraction (indicated in Figure 33 by a container surrounding the blocks). Similar to the streaming case described in the previous chapter, the extraction algorithm consists of three components:

1. Computation of the quality contribution information of each layer in the bitstream. This is carried out as a post-encoding operation.

2. Signaling of this information in the bitstream.

3. Extraction based on this quality metadata at an intermediate network node.

Now, we describe our algorithm for each of these components in detail. First we look at the computation of the quality information. Then, we describe how metadata information is signalled in the stream and finally we propose the extraction algorithm that uses this quality metadata information along with paired-frame extraction.

4.3.3.1 Computation of quality metadata information

The computation of quality metadata information is essential for making content dependent extraction decisions. The quantities computed are shown in Figure 34. This is similar to the metadata information computed for the one-way streaming case as illustrated in Figure 12. The metadata computation process is invoked upon receiving every frame unlike the one-way streaming case where it was computed once the entire GOP was received. This is done to avoid additional end-to-end delays that would be incurred if the extractor had to wait to receive frames which arrive only every δ. For every frame of data received, a total of four decodings are performed (indicated by `bq,mq,cq` and `dq` in Figure 34). They are at the lowest and highest quality layers of the current frame, each predicted from the lowest and highest quality layers of their parent. Next, the distortions are computed for each of the four types of extracted frames with respect to the highest quality reconstructed stream with all the

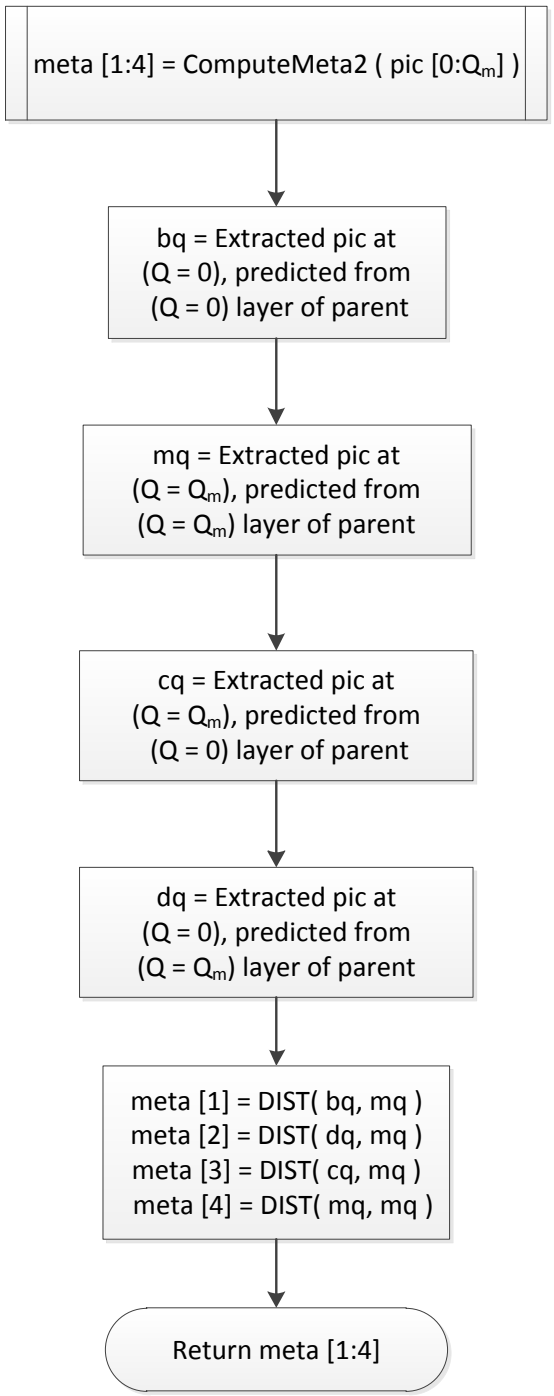


Figure 34: Computation of metadata needed for paired-frame extraction.

enhancement layers. These distortion values are stored as the metadata information for the current frame. These four basic decodings are essential to evaluate the quality contributions of the lowest and highest quality layers in the frame, which in turn depends on the number of layers reconstructed for their parents. It is very important to reduce the number of decodings since post-processing needs to be done in real time and interactive applications are limited by tight constraints on end-to-end delays. Hence, the quality contributions of the in between MGS layers are not computed but they are predicted according to their sizes using the procedure described for one-way streaming.

4.3.3.2 Signaling of quality metadata information

The quality metadata information for the current frame is transmitted as part of the SVC bitstream either in the *priority_id* field of the NAL unit header or in separate SEI messages. Hence, the extractor can access this information without having to decode the stream.

4.3.3.3 Paired-frame extraction using quality metadata information

Using the quality information that has been embedded into the SVC bitstream for each frame, it is now possible to improve the paired-frame technique of extraction by making content-dependent decisions. The paired-frame technique gave preference to temporally more important frame in the pair by allowing it to exceed its allocated bit budget and consume extra bits from the second frame in the pair, which is at a higher and lesser important temporal layer. This preference was given in a content independent way for every first frame in every pair. We modify this preference by using the quality metadata information. The proposed paired-frame extraction using quality information is illustrated as a flow chart in Figure 35. When a frame arrives at the MCU, the extractor checks its temporal ID to see whether it is the first or second frame in the pair. For the first frame ($T < T_m$), it invokes the regular paired-frame

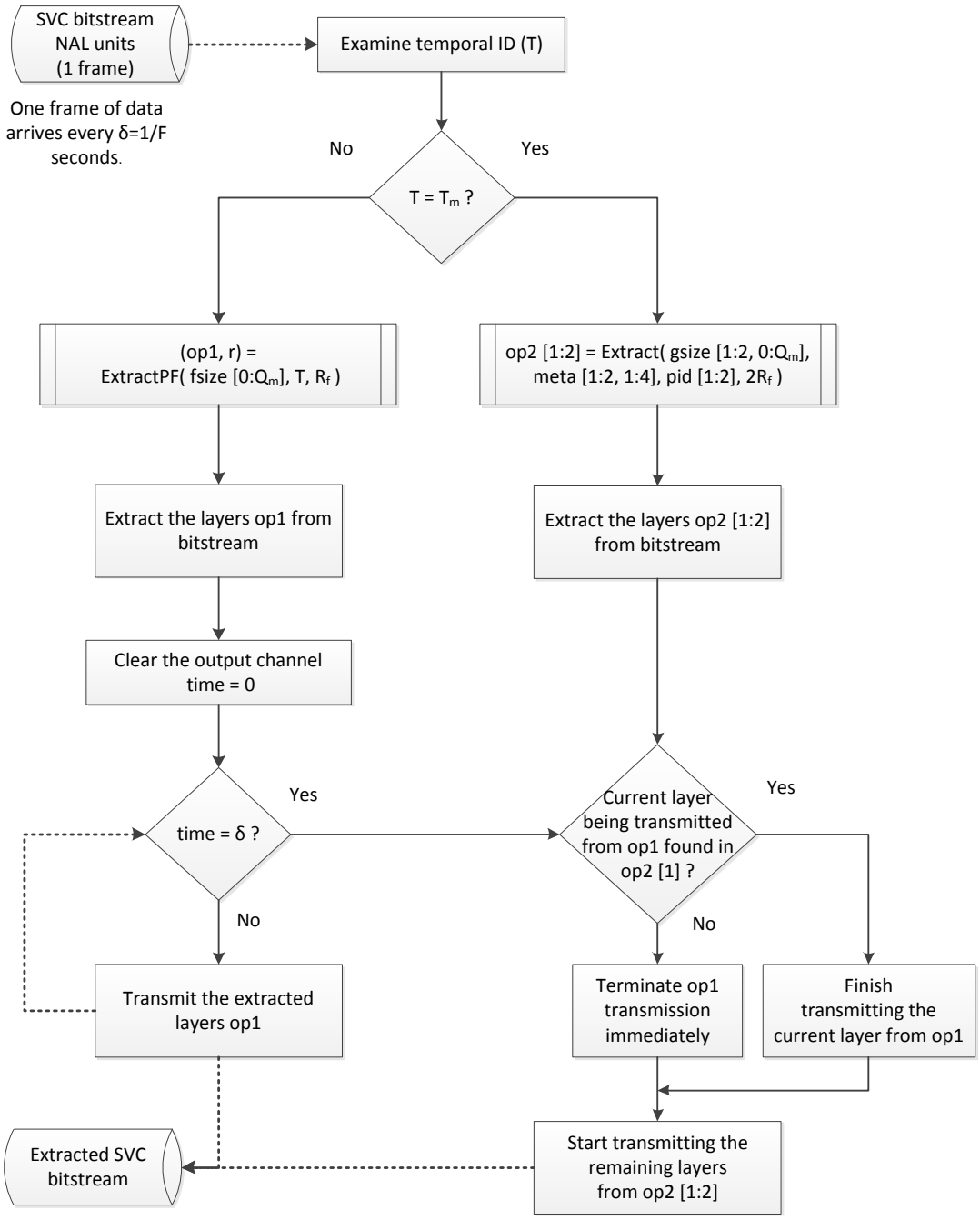


Figure 35: Flowchart for paired-frame extraction using quality information.

extraction that is illustrated in Figure 31. The function returns the list `op1`, which is the list of layers to be extracted for the current frame. The layers are extracted according to this list and placed on the output channel.

At time δ later, the next frame, i.e., the second frame in the pair arrives at the MCU. By checking its temporal layer ID, the extractor identifies it to be the second frame in the pair and then it invokes the `Extract()` function, which was defined earlier for one-way in Figure 14. The input to this function is both the current frame (second frame in the pair) received at `time` $= \delta$ and the previous frame (first frame in the pair) received at `time` $= 0$ and which is currently being transmitted in the output channel. The metadata information received for the current frame as well as the previous frame in the pair is also used by the extract function. The bit budget input is the total budget allocated to the pair ($2R_f$). The priority IDs input are simply 1 for the first frame and 2 for the second frame in the pair. The `Extract()` function outputs (`op2`) the RD optimal decision of layers to be extracted for both the frames in the pair. But it must be remembered that the first frame in the pair (which was received at `time` $= 0$) has already been extracted and transmission of the extracted layers has begun. Hence, at `time` $= \delta$, the transmission of the previously extracted frame is stopped for a moment and the output (`op2`) of the `Extract()` function is checked. If the current layer of the previous frame (the pair's first frame) that is being transmitted is included in this list, then its transmission is continued till that quality layer is fully transmitted. Then, the transmission of further layers of the two frames in the pair continue according to the list `op2`. If the current layer of the previous frame in the pair is not included in the RD optimal extracted list `op2`, then the current transmission is ended abruptly and new transmission begins according to the layers specified in the list `op2`.

In this way, the consumption of additional bits by quality layers in the first frame

in the pair is controlled in an RD optimal way, i.e., these layers are extracted depending on their contribution to the overall reconstructed video quality. The proposed extractor uses distortion metadata information of each quality layer and estimates the reduction in distortion obtained on decoding a specific quality layer. Then, it extracts the quality layers among the two frames in the pair in decreasing order of their quality contribution. If the extractor identifies that the quality layer in the first frame that is responsible for extra consumption of bits over the allocated limit of R_f is not contributing more to reducing the distortion than a quality layer in the second frame, it would not extract that quality layer from the first frame. So, by checking the output list `op2` at `time` $= \delta$ during the midst of the transmission of the first frame in the pair, it is possible to avoid the extra consumption of bits in case it does not contribute high enough to the reconstructed video quality when compared to the layers in second frame of the pair. The jitter and end-to-end delay performance of this technique is identical to that of the paired-frame extraction proposed in the previous subsection. This also requires a jitter compensation buffer of δ at the receiver and the end-to-end delay goes up by δ when compared to single frame extraction. This is usually not a problem since most receivers have dejitter buffer of 2δ or more and the increase in end-to-end delay is acceptable as long as it is within the specified limits of interactive communication.

4.4 Experiments and Results

In this section, we validate the proposed algorithms of the previous section through experiments and results. We quantify the performance improvements obtained by using the paired-frame extraction using quality information over regular paired-frame extraction and frame-by-frame extraction. First, we describe our conversational video sequences database. This is followed by the video quality results at various bitrates for each of the three extraction techniques. Paired-frame extraction using quality

information shows a maximum quality increase of about 0.2 dB when compared with simple paired-frame extraction and a quality increase of about 1.3 dB when compared with frame-by-frame extraction. Finally we show a snapshot of each of the algorithm's performance and a few sample frames that show the superiority of paired-frame extraction with quality information over the other two techniques.

4.4.1 Conversational Sequences Database

In this subsection, we describe the conversational video sequences database. Since there are no standard conversational sequences for testing, we shot some sequences at 720P and 25 fps. These sequences are labeled as RP1, RP2, RP3 and RP4. They represent actual video conferencing scenarios in well-lit conference rooms. The details of these sequences along with the various encoding parameters used are described in Table 10. The scan type is progressive in YUV 4:2:0 format. The frame rate is 25 fps for all the sequences. The GOP structure uses only P pictures, as shown in Figure 28, to ensure a zero-delay encoding structure. With a GOP size of 8 frames, the number of GOPs used is 50 for all the four sequences. All the sequences include an additional frame in the beginning, which is encoded as an IDR picture. All the sequences are encoded using the JSVM SVC encoder [70], which is the reference software issued by ITU–T. The quantization parameters (QP) used for encoding are also shown in Table 10. Since the GOP size used is 8, there are 4 temporal layers in all the sets. For simplicity, only one spatial layer with 6 quality layers (including the base quality layer) is used. Sample frames from all the sequences are shown in Figure 36. The encoded bitrate of each layer for all the sequences is shown in Table 11. From the table, we can see that we cover a wide range of bitrates, from 100 kb/s to 2000 kb/s.

Table 10: Conversational sequences' characteristics and encoding parameters.

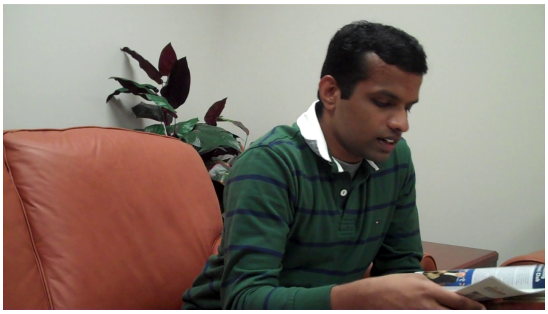
Parameter	Value
# of sequences	4
Sequence names	RP1, RP2, RP3, RP4
Spatial resolution	1280×720 (720p)
Scan type	Progressive
YUV format	4:2:0
Frame rate	25 fps
# of frames	401
Duration	16.04 s
GOP size (N)	8 frames
Sequence structure	IDR–{P3-P2-P3-P1-P3-P2-P3-P0}× 50
Base layer QP	40
MGS layer QP	30
# of Temporal layers	4 ($T = 0, 1, 2, 3$)
# of Quality layers	6 ($Q = 0, 1, 2, 3, 4, 5$)
# of Spatial layers	1 ($D = 0$)

Table 11: Bitrates (kb/s) of the SVC encoded sequences in the conversational test sequences database (720p).

Layers (D, T, Q)	RP1	RP2	RP3	RP4
(0,0,0)	133.40	76.60	87.50	85.70
(0,1,0)	185.90	102.00	121.60	122.40
(0,2,0)	246.80	131.20	167.30	167.00
(0,3,0)	314.00	164.30	222.90	220.40
(0,0,1)	701.50	645.90	462.00	444.40
(0,0,2)	892.00	827.30	555.40	528.10
(0,0,3)	984.50	920.70	603.70	571.30
(0,0,4)	1042.60	977.00	635.60	599.50
(0,0,5)	1060.20	996.10	648.60	611.00
(0,1,1)	892.80	750.30	580.70	569.00
(0,1,2)	1113.60	952.00	693.80	670.60
(0,1,3)	1221.90	1057.70	753.20	724.20
(0,1,4)	1291.10	1123.80	794.10	760.30
(0,1,5)	1316.70	1150.70	813.90	778.00
(0,2,1)	1112.20	862.30	730.50	717.10
(0,2,2)	1364.10	1084.80	868.30	840.00
(0,2,3)	1490.40	1204.80	943.10	907.60
(0,2,4)	1573.80	1283.50	997.90	956.00
(0,2,5)	1611.90	1322.00	1029.70	985.10
(0,3,1)	1358.60	987.60	910.70	891.50
(0,3,2)	1648.00	1236.10	1080.80	1043.50
(0,3,3)	1801.00	1376.80	1179.60	1133.80
(0,3,4)	1907.00	1475.00	1257.40	1203.90
(0,3,5)	1967.00	1532.30	1310.90	1253.90

(a) RP1

(b) RP2



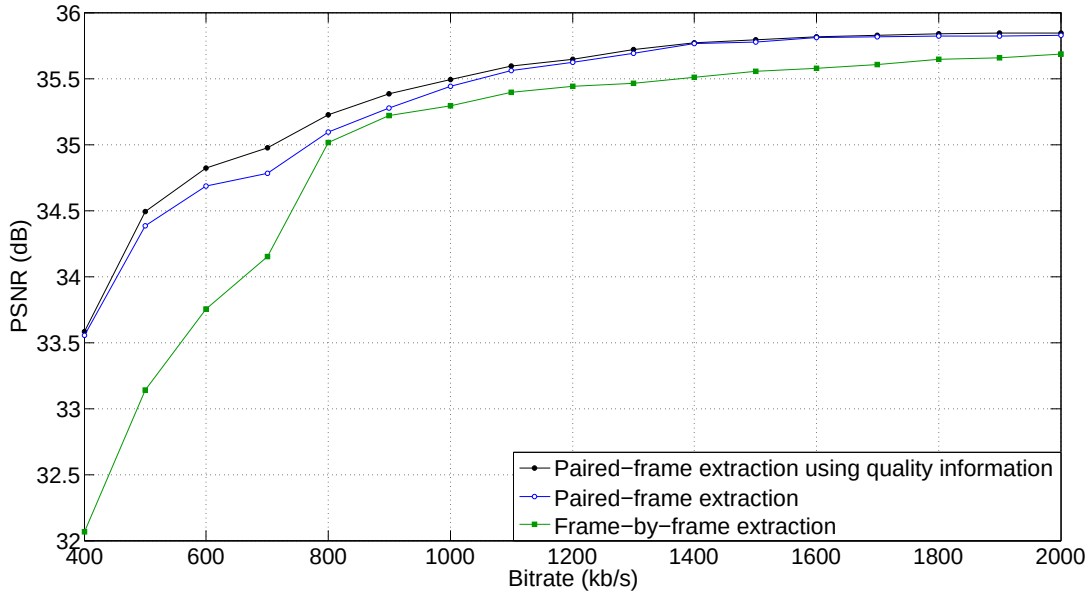
(c) RP3

(d) RP4

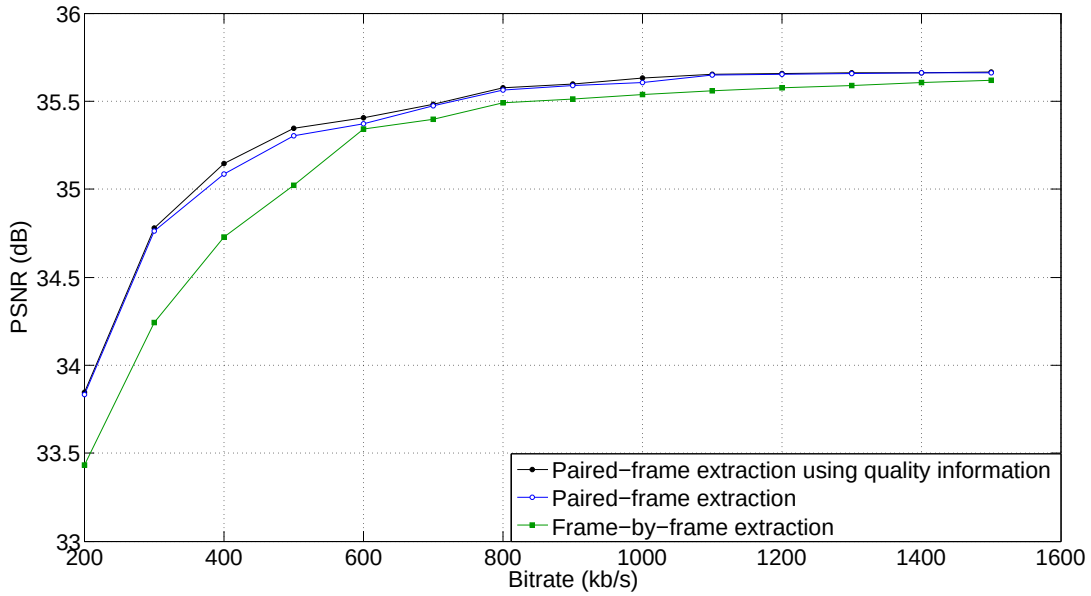
Figure 36: Sample frames from the conversational video database (720p).

4.4.2 Video Quality Evaluation

In this section, we describe the experiments and results obtained for the quality of decoded video that has been extracted at various bitrates corresponding to the available bandwidth values in the channel. Extraction is performed using all the three techniques namely, frame-by-frame, paired-frame, and paired-frame using quality information. The quality of decoded video is measured using PSNR, a full-reference metric using the fully reconstructed video (with all MGS quality layers) as the source reference. This approach is correct since we are interested in comparing the rate distortion performance of the various extraction techniques. When all layers are extracted, maximum performance is reached. An extraction technique cannot perform better than extracting all the layers. Hence, maximum quality reconstruction is used as a reference.

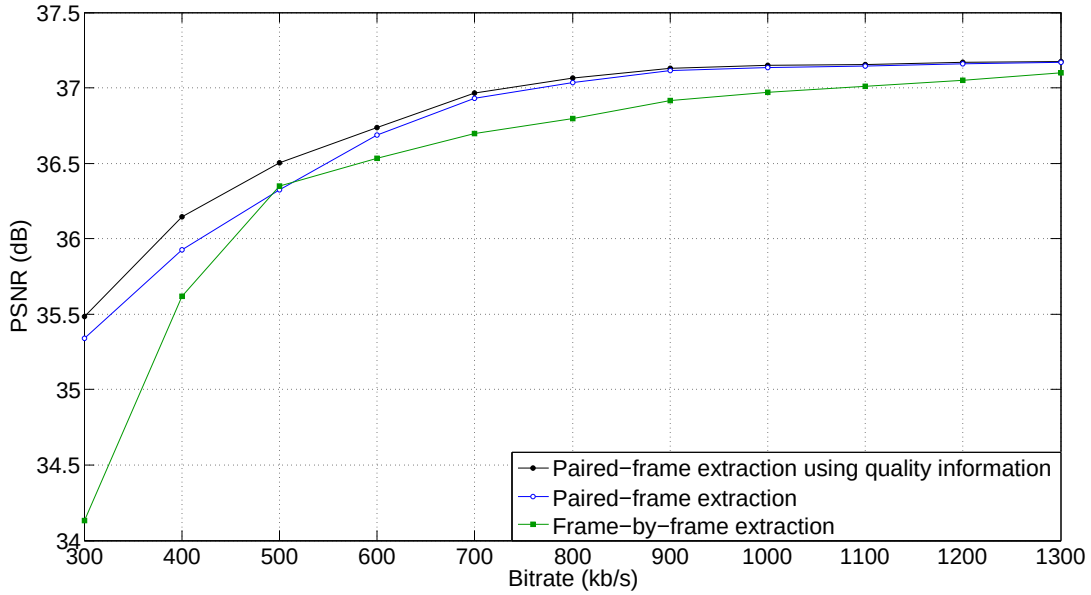


(a) RP1

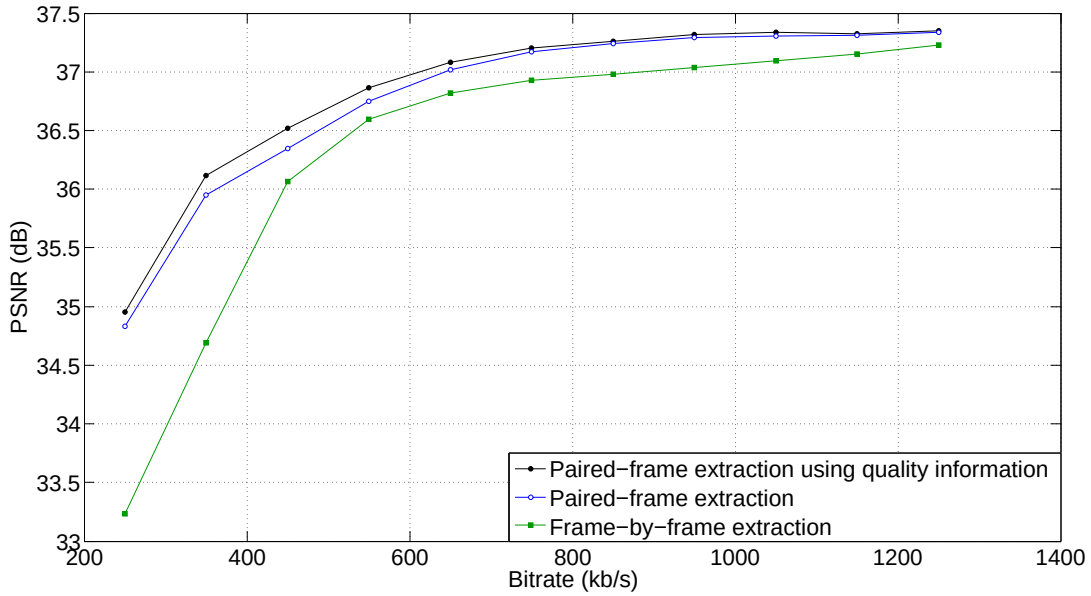


(b) RP2

Figure 37: Video quality vs. bitrate (available bandwidth) for RP1 and RP2.



(a) RP3



(b) RP4

Figure 38: Video quality vs. bitrate (available bandwidth) for RP3 and RP4.

Figures 37 and 38 plot the decoded video quality (PSNR) with respect to variations in bitrates used for extraction (i.e. the available bandwidth in the channel) for the sequences RP1, RP2, RP3 and RP4. Each plot shows extraction using frame-by-frame, paired-frame, and paired-frame using quality information. For all the sequences, paired-frame extraction using quality information has higher reconstructed video quality when compared to the other two techniques. This is because of the content-dependent, rate-distortion optimal extraction decisions taken by the paired-frame extraction technique using quality information. The frame-by-frame extraction technique performs the worst since extraction decisions are made over a window of one frame and hence, temporal importance among frames plays no role in the decisions. When the available bit budget is not sufficient for extracting even the base quality layer of a frame, then the frame is skipped. This has serious consequences on the extraction of future frames in the GOP that are dependent on this frame for prediction. Since the parent frame was skipped, the child frames at higher temporal layers cannot be extracted even when the available bit budget allows their extraction. The paired-frame extraction performs better than frame-by-frame extraction since the decision window comprises of a pair of frames, the first frame at a lower temporal layer (more important) and the second frame at the maximum temporal layer (least important). The extractor allows the first frame to overshoot its bit budget by consuming extra bits from the second frame, which is less important from a prediction standpoint. Hence, more quality layers are extracted for the first frame than the second frame. This improves the quality of the first frame directly and also improves the quality of the second frame indirectly since it is predicted from the first frame. Moreover, this temporal layer preference assures that at least the base quality layer of the temporally more important frame (first frame in the pair) is always extracted so that future child frames at higher temporal layers that are dependent on this frame

can be extracted at a later time as long as the bit budget allows it. However, the preference to lower temporal layer is done universally, in a content independent fashion, for every first frame in every pair. This contributes to its reduced performance when compared to paired-frame extraction using quality information. Quality metadata information about each frame helps the extractor evaluate the quality contribution of each layer in the frame pair and these are reflected in the extraction decisions made over the two-frame window. Allowing a quality layer in the first frame of the pair (temporally more important) to consume extra bits originally allocated for the second frame (temporally less important) is done only when that quality layer reduces the distortion more than the competing quality layer from the second frame. Such decisions help in the extraction of only those layers that contribute maximally to the reduction of distortion in the extracted video and hence, increase the reconstructed video quality.

From the figure, we notice that at higher bitrates, paired-frame extraction with or without using quality information has similar reconstructed video quality. This is because of the fact that at such bitrates, most of the quality layers are extracted for both the extraction types and hence, they perform in a similar manner. However, even at higher bitrates, frame-by-frame technique has reduced quality due to the fact that the base quality layer of some frames could be large enough such that it may not fit within the available bit budget for that frame and hence, it is skipped. This adversely affects all the following frames that depend on it since they cannot be extracted in spite of available bit budget since dependency conditions would not have been satisfied.

Table 12 shows the maximum increase in PSNR obtained for the paired-frame extraction technique using quality information when compared with simple paired-frame extraction and frame-by-frame extraction techniques. When compared to simple paired-frame extraction, the increase is about 0.2 dB averaged over all sequences,

Table 12: Max. increase in PSNR (dB) obtained for paired-frame extraction using quality information when compared with paired-frame extraction and frame-by-frame extraction for conversational sequences.

Sequence	Max. increase over paired-frame (dB)	Max. increase over frame-by-frame (dB)
RP1	0.20	1.52
RP2	0.06	0.54
RP3	0.22	1.35
RP4	0.17	1.72

with a maximum increase of 0.22 dB for RP3 sequence. Compared to frame-by-frame technique, the maximum increase is about 1.3 dB averaged over all sequences, with a maximum increase of 1.72 dB for RP4 sequence. This is understandable since both paired-frame with quality information and simple paired-frame based extractions use a decision window of two frames. They differ only in the aspect of assigning priority among the various quality layers in the adjacent frames. However, frame-by-frame technique assigns no temporal priority and hence, has a much reduced performance.

4.4.3 Snapshot of Algorithms' Performance

Figure 39 shows a snapshot of the performance of all the three extraction techniques using a set of 100 frames from RP3 sequence extracted at 1300 kb/s and RP4 sequence extracted at 1250 kb/s as examples. We clearly see from the plots that paired-frame extraction using quality information always performs equal or better than the other two techniques. Frame-by-frame extraction has a number of dips in the plot. This is because of its decision window of one frame which sets a narrow limit on allocated bit budget for a frame. When the size of the base quality layer of the frame is too big to be extracted, it is skipped resulting in poor quality of that frame and all the frames that are dependent on it (they cannot be extracted too, since they would not satisfy

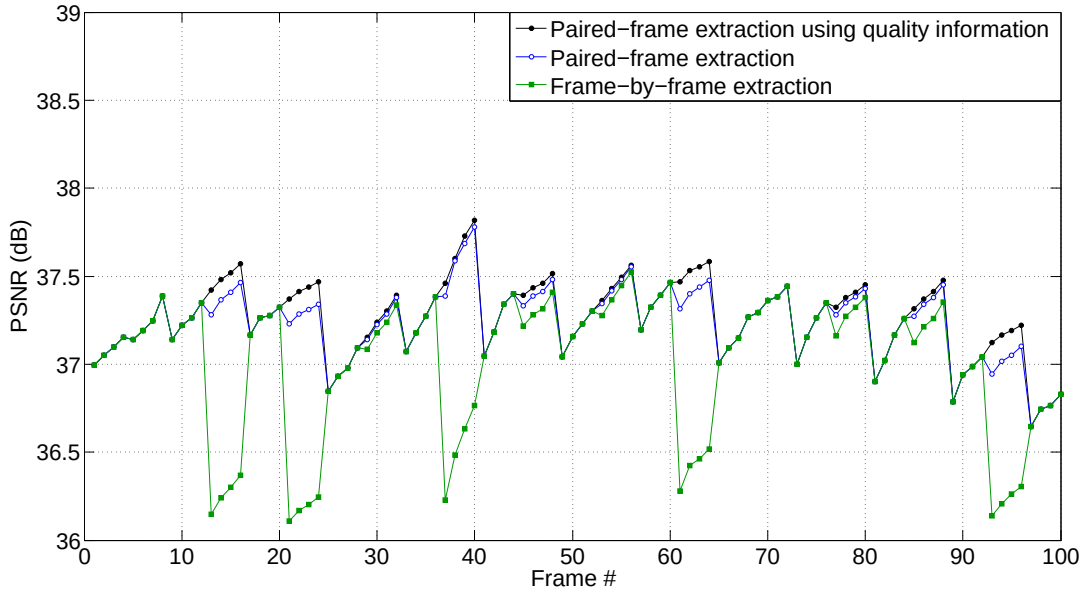
the dependency constraints). We can notice that the paired-frame extraction has slightly lower or equal quality than paired-frame extraction with quality information. This is because of the fact that both extraction techniques use a decision window of two frames and differ only in the way of handling priority among the various quality layers in each frame pair.

4.4.4 Sample Frames

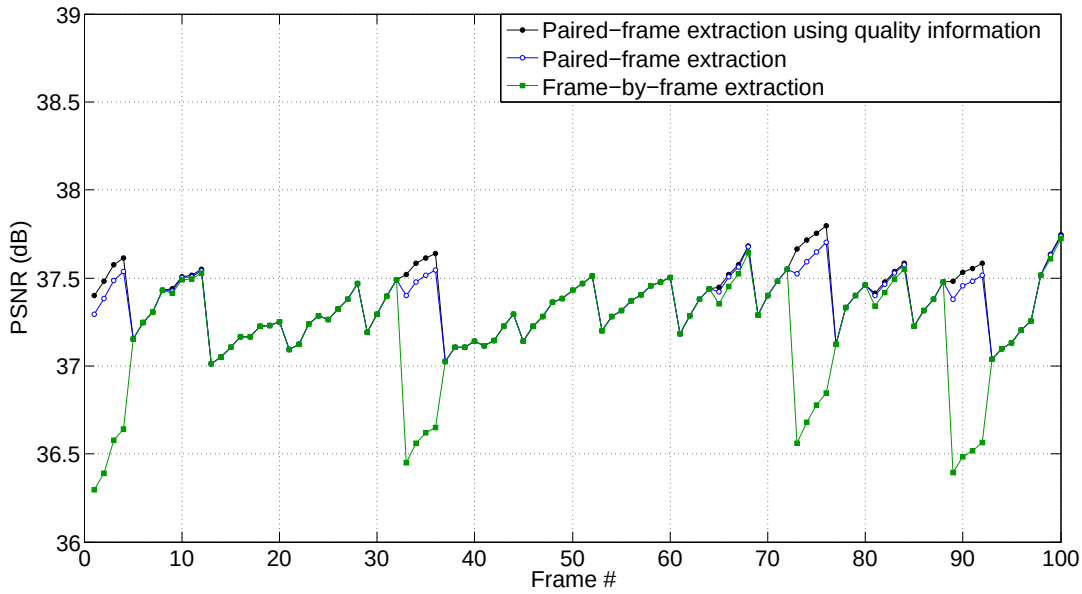
In this subsection, we show a few sample frames from the test sequences to show the visibility of artifacts among the video streams extracted using the various extraction algorithms. Figure 40 shows Frame # 344 of RP1 sequence extracted at 700 kb/s using paired-frame with quality information and using frame-by-frame technique. The frame extracted using paired-frame with quality information has a smoother feel with lesser blockiness artifacts when compared to the frame extracted using frame-by-frame method. This effect is visible in the face and chest regions that are marked in the figure. Similarly, Figure 41 shows Frame # 184 of RP2 sequence extracted at 1400 kb/s using the same two techniques. Again, the face and chest regions have more blockiness in the frame extracted using frame-by-frame method. Finally, Figure 42 shows Frame # 304 of RP4 sequence extracted at 1250 kb/s. The eye and upper-arm regions have visible blockiness in the frame extracted using frame-by-frame technique.

4.5 Summary

This chapter has focused on solving the problem of maximizing the video quality of SVC-encoded content under varying bandwidth conditions in a real-time video conferencing scenario. Interactive applications are characterized by tight end-to-end delay and jitter constraints, which pose special challenges in designing SVC bitstream extraction algorithms for such applications. In this chapter, we have explored enterprise video conferencing as an application scenario. We have proposed an end-to-end architecture using an SVC-based multipoint control unit (MCU) that extracts the



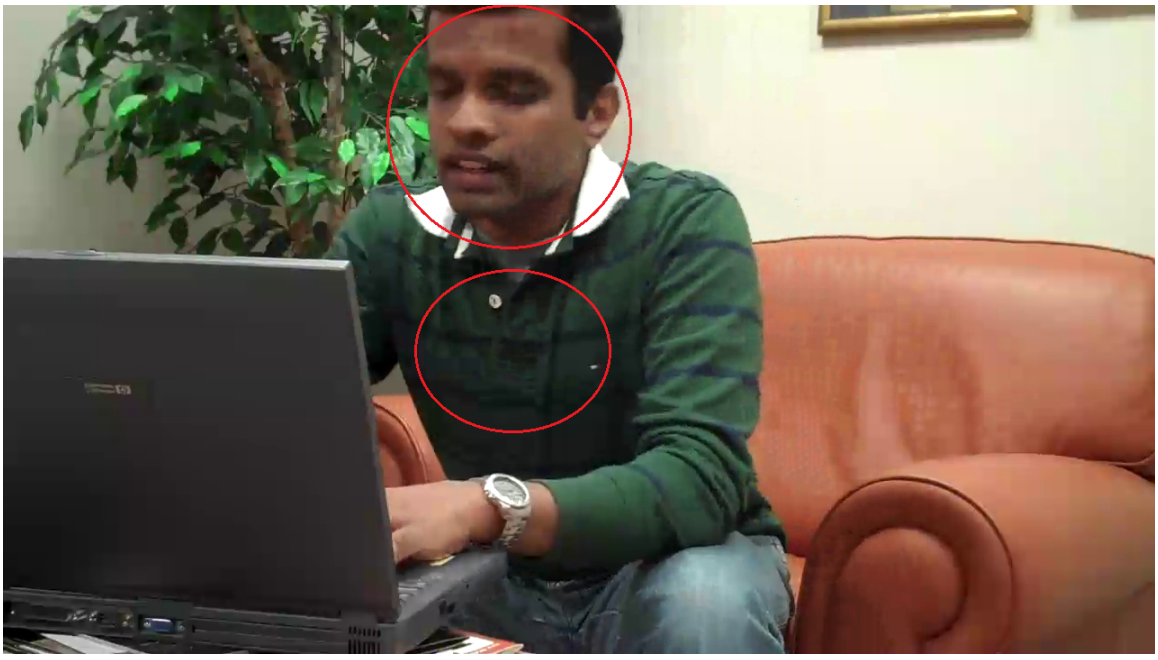
(a) RP3 extracted at 1300 kb/s



(b) RP4 extracted at 1250 kb/s

Figure 39: Quality (PSNR) of a set of 100 frames extracted using all the three extraction techniques.

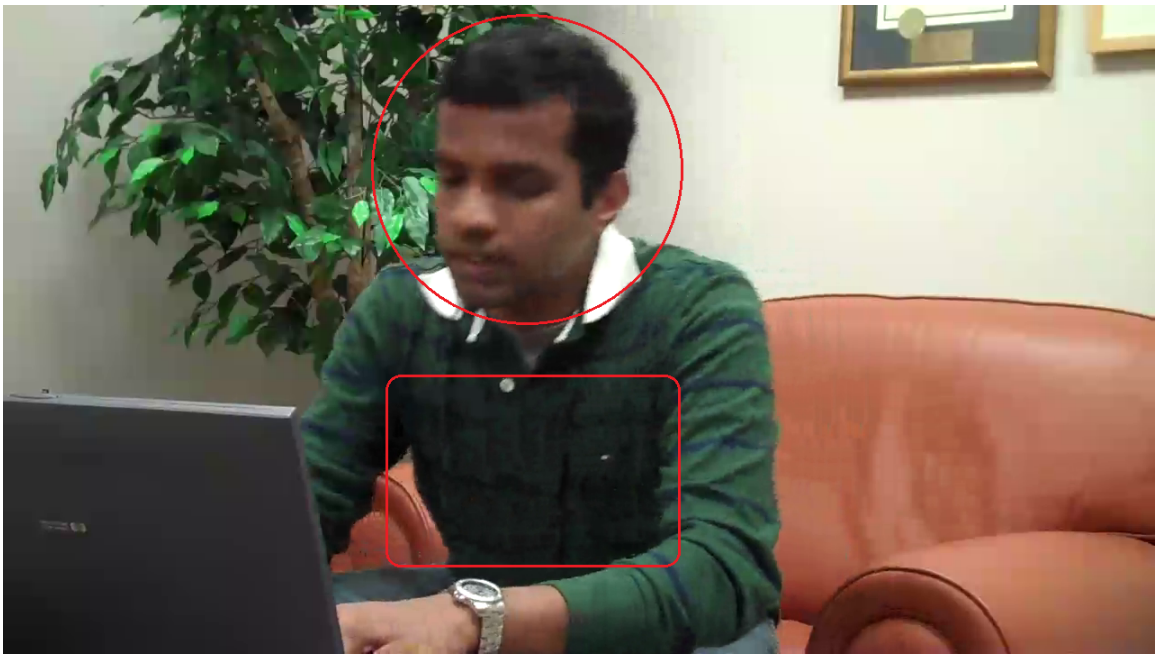
(a) Extracted using paired-frame using quality information



(b) Extracted using frame-by-frame

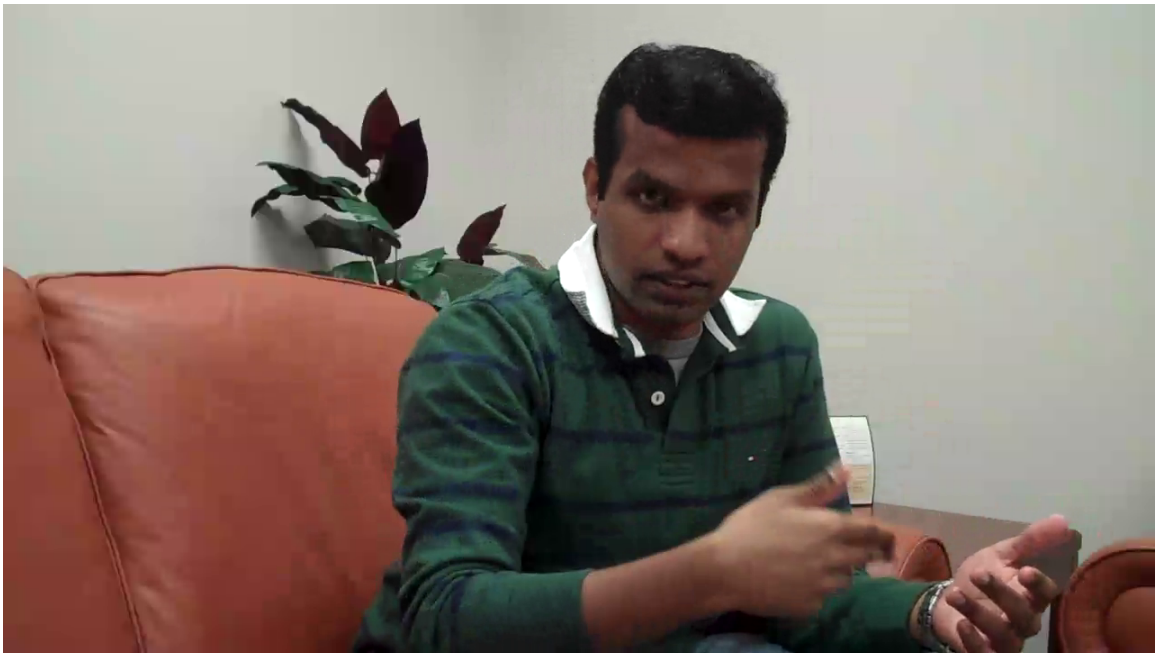
Figure 40: Frame # 344 of RP1 sequence extracted at 700 kb/s.

(a) Extracted using paired-frame using quality information

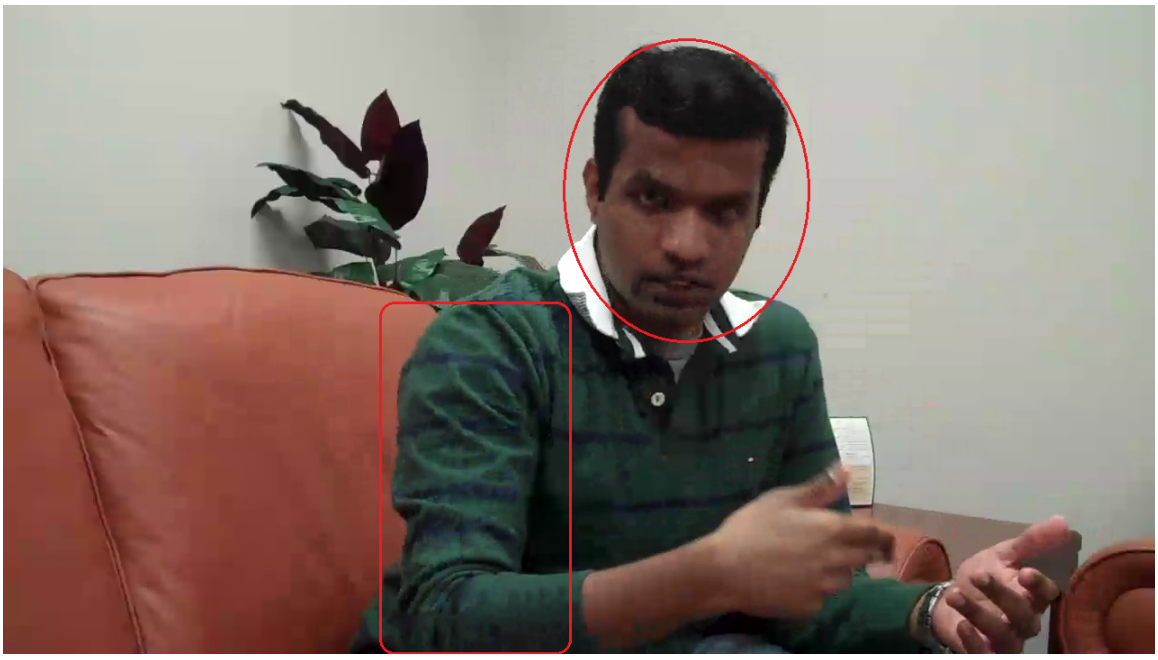


(b) Extracted using frame-by-frame

Figure 41: Frame # 184 of RP2 sequence extracted at 1400 kb/s.



(a) Extracted using paired-frame using quality information



(b) Extracted using frame-by-frame

Figure 42: Frame # 304 of RP4 sequence extracted at 1250 kb/s.

bitstreams and coordinates the flow of traffic between the multiple parties involved in such a communication.

Using this application as the motivation, we have performed an end-to-end delay analysis of the various delay components in such a real-time interactive communication system. The role of jitter compensation buffer in smoothing the variations in arrival times of the data frames due to varying bandwidth conditions have been studied. The differences between one-way streaming and two-way conferencing on the areas of content type, encoding complexity, GOP structure, etc., have been discussed. The problem of RD optimal SVC bitstream extraction has been proposed in a general framework with a decision window of n frames at the extractor. Depending on the tolerable limits of end-to-end delay and jitter, n can vary from 1 to size of the GOP (N).

Solutions with extraction decision windows of one and two frames have been proposed to the above problem. Windows larger than this size place heavy demands on jitter compensation at the receive buffer and increase the end-to-delay beyond acceptable limits for interactive communication and hence, have not been considered. Frame-by-frame extraction uses a decision window of one frame and extracts the quality layers starting from the base quality layer of that frame. Extraction stops when the allocated bit budget for that frame is reached. It ensures a jitter-free transmission and hence, it does not require a jitter compensation buffer at the receiver. However, the reconstructed video quality is poor as it treats each frame equally, independent of their temporal importance. When the base quality layer of a frame is too big to be sent within the bit budget, it is simply skipped. This problem is solved by using a paired-frame technique of extraction which uses a decision window of two frames that are located adjacent to each other in display order and belong to different temporal layers. The first frame in the pair is at a lower temporal layer (more important) and the second frame is at a higher temporal layer (less important). Hence, the extractor

gives more preference to the first frame and allows it to consume extra bits than the allocated bit budget for the frame. These extra bits come from the allocation for the second frame in the pair. This technique favors the temporally more important frame and tries to ensure that at least the lowest quality layer of that frame is extracted. Hence, this extraction produces better reconstructed video quality when compared to frame-by-frame extraction. This technique requires jitter compensation equal to one frame interval at the receive buffer. Since this technique favors the first frame of every pair universally, the decision is content independent and hence, not RD optimal. This leads to less optimal extraction but still better than frame-by-frame extraction.

The RD optimality of the paired-frame extraction is improved by making extraction decisions based on the contribution of each layer in the frame pair towards reconstructed video quality. This requires the computation of meta data quality information, which is carried out as a post-encoding process for each frame. Based on this information, the extractor makes RD optimal decisions. This technique is called paired-frame extraction using quality information. Quality metadata information about each frame helps the extractor evaluate the quality contribution of each layer in the frame pair and these are reflected in the extraction decisions made over the two frame window. Allowing a quality layer in the first frame of the pair (temporally more important) to consume extra bits originally allocated for the second frame (temporally less important) is done only when that quality layer reduces the distortion more than the competing quality layer from the second frame.

Experiments and results show the superiority of paired-frame extraction using quality information when compared to the other two techniques. It shows a maximum quality increase of about 0.2 dB when compared with simple paired-frame extraction and a quality increase of about 1.3 dB when compared with frame-by-frame extraction. Sample frames show that perceptual quality is better with paired-frame extraction using quality information when compared to frame-by-frame extraction.

CHAPTER V

SVC BITSTREAM EXTRACTION FOR 3DTV

Stereoscopic 3DTV [80–83] has already been launched to homes over cable and satellite networks. Recently, the Masters Tournament was broadcast by CBS and Comcast in side-by-side, frame-compatible MPEG-2 [84] format to homes with HDTV settops, and it was also streamed live on the Internet in 3D. In our work, we extend the streaming application using SVC to 3D content. Except for the content being 3D, the basic principles of one-way streaming hold good here. However, 3DTV technology is characterized by multiple differences in content format (frame-compatible and full-resolution stereo) and display technologies (active shutter and passive polarization). When viewing 3D content, humans perceive depth by the cognitive processing of two different perspectives, one for left eye and the other for right eye, of the same scene. Hence, proper perception of depth is maintained by perfect synchronization between the left-eye and the right-eye views. In 2D videos, the end-user experience is objectively analyzed by measuring the reconstructed video quality using standard distortion measures like mean square error (PSNR). However, in 3D, the user QoE cannot be adequately evaluated with simple distortion measures because of the added dimension of depth perception and the individual preferences of each user. End-user 3D experience varies widely among users depending on their age since glasses must be worn at all times when viewing such content. Prolonged viewing causes visual fatigue in some users because of the additional cognitive processing required in perceiving depth [85]. Hence, subjective tests are important in analyzing the user QoE when watching 3D content [86,87]. From the network perspective, additional bandwidth is required for streaming of 3D content, especially when full-resolution stereo formats

are used, where two bitstreams are delivered (one for the left eye and one for the right eye). Such heterogeneities make 3D content different from their 2D counterparts, thus making the live streaming of such content more challenging.

5.1 3DTV – Content Formats and Displays

A variety of 3D representation formats are in use today, the key ones being full-resolution stereo, stereo interleaving (commonly known as frame-compatible format), and 2D plus depth [22,88]. Full resolution stereo employs two bitstreams at full frame rate: one for the left-eye view and another for the right-eye view. Hence, each eye receives the complete bitstream at the original frame rate. For real-time encoding, it is necessary to keep the encoding complexity and processing delay at a minimal level. This leads to independent encodings of the left and right-eye views. The disadvantage with such encoding is that the bandwidth required is twice that of regular streaming since the similarities between the views is not exploited. For N-way multiview, the bandwidth increases N-fold. This bandwidth requirement can be reduced by efficiently compressing the multiple views while exploiting the redundancies between them. The pictures are predicted both temporally and spatially from adjacent views. This is called inter-view prediction and is used in multi view coding (MVC), a recently standardized extension of H.264/AVC [45, 89]. However, this increases the complexity of the encoding process and hence, it might not be suited for real-time encoding purposes (e.g., broadcasting live sport events). It can still be used for offline encoding of stored content like 3D movies.

The most predominant format in current use is the stereo-interleaved format, also known as the frame-compatible format. Here, the left-eye and right-eye views are subsampled and interleaved into a single frame. There are a number of ways of interleaving the two views including side-by-side, top-bottom, row-interleaved, column-interleaved, checkerboard, etc. In time multiplexing, the left-eye and right-eye views

are interleaved as alternating frames. The main advantage in these interleaved formats is that existing infrastructure can be used for the distribution and delivery of 3D content since there are no additional bandwidth and codec requirements. The streams can be encoded and decoded by existing codecs. Hence, all the cable and satellite service providers currently broadcast 3D content in this format [90–92]. However, this convenience comes at the price of losing half the content information along the spatial or temporal dimension, which affects the quality of the delivered 3D video. The other less known format is the 2D plus depth format. Here, the regular 2D video is streamed and the depth map information is sent as an auxiliary video or supplemental information. It provides backward compatibility with legacy 2D decoders so that they can simply ignore the depth information and display the regular 2D content. The only drawback in this format is that the depth range is very limited and not well suited for achieving high user QoE.

The common display technologies used in 3DTV today [93] include active shutter and passive polarized systems. In an active shutter system, the left-eye and right-eye views alternate in time when shown to the user. Each view has full spatial resolution. The resulting frame rate is twice the frame rate of the individual left-eye and right-eye bitstreams. The total brightness is reduced by a factor of two since one of the two eyes is always shut at any given point of time (achieved through the active shutter glasses). The viewing angle is wide and the perceived depth is intense and lively. However, some people experience visual fatigue like eye strain and mild headache after watching for around 10-15 minutes. Hence, it is highly suitable for short duration and high action sport sequences, video games, etc. Existing displays can be used for this system but it requires active shutter glasses and a synchronizing unit (to match the shuttering rate with the refresh rate), which is expensive.

The other most common display technology is the passive polarized system. Here the left-eye and the right-eye views are interlaced onto a single frame. By using

polarizing filters, each alternate row on the display screen is left- and right-circularly polarized. The glasses also use similar polarizing filters. Hence, each eye receives every other row of video information. The frame rate remains the same as that of the bitstream. There is no reduction in brightness since both the eyes are open at all times. The sense of depth perceived is subtle and the viewing angle is narrow. Such displays are soft on the eyes and much suited for prolonged viewing, for e.g., a movie. This system requires specialized 3D monitors with every row alternately polarized. However, the glasses are simple filters and hence very cheap.

The less common type of 3D display technology includes autostereoscopic displays [94]. These displays do not require glasses to view the 3D content. Such displays come in various forms such as lenticular types, parallel barrier types, etc.

In our work on live streaming 3D content over the Internet, we use SVC-based independent encoding and simulcasting of each view in full-resolution stereo format. This minimizes the encoding delay and allows bitrate adaptations at intermediate network nodes so that the overall perceived quality can be optimized with respect to varying bandwidth conditions in the channel. However, our extraction algorithms developed for SVC-based streaming can be readily applied to the scalable version of MVC (known as SMVC [23, 95]) since both the encoding techniques share the same features that are needed for bitrate adaptation and layer extraction algorithms.

5.2 Streaming of 3D Content – Architecture and Algorithms

Streaming of 3D content in frame-compatible format has the disadvantages of loosing half the resolution in either horizontal or vertical directions. This leads to overall poor quality of experience (QoE) for the end user. One of the solutions to boost the end user QoE is to stream 3D content in full-resolution stereo, i.e., the left and right-eye views are transmitted in full-resolution. Streaming to end users in a variety of heterogeneous network environments requires the adaptation of each of the left

and right-eye views according to the current available bandwidth conditions in the channel. The end-user QoE depends on the final perceived quality in 3D when both the left and right eye views are decoded and displayed either in a passive polarized or an active shutter system. Hence, the main challenge in 3D streaming is in performing the adaptation of the left and right eye views to changes in available bandwidth conditions so that the perceived end-user quality is maximized.

5.2.1 Encoding Left-eye and Right-eye Views

As we have seen in the previous chapters, SVC-based encoding offers an ideal solution for encoding videos when content adaptation is required. Encoding the left and right-eye views into a number of spatial, quality and temporal layers ensures scalability along each dimension and gives flexibility to the extractor in performing an RD optimal bitstream extraction. Independent encoding of the left and right-eye views ensures that each bitstream can be extracted independently and hence, an optimal extraction strategy can be developed for each view. This contributes to the improvement in overall QoE. Moreover, in applications such as live broadcast of events in 3D (e.g. sports events), real-time encoding is essential. Independent encoding reduces the encoding complexity, and hence the encoding time, since both the views can be encoded in parallel. The only disadvantage of independent encoding is the reduction in overall compression efficiency.

The compression efficiency can be improved by making one of the two views independent and the second view dependent on the first view. The dependent view is predicted from the independent view using inter-view prediction mechanisms (multi-view video coding – MVC). One of the main disadvantages of this technique is that inter-view prediction is very computation intensive and may not be achievable in real time. Encoding of each view cannot be parallelized since the views are dependent on each other. This leads to sequential encoding and thus increases the encoding time.

This limits the usage of MVC in applications such as live 3D broadcast etc. However, it could be used for on-demand content when the video is pre-encoded and stored. MVC in its original form does not have any scalable features. It simply uses inter-view predictions to predict one view from the other and multiplexes the two views into one bit stream. Using such an MVC encoded video for streaming applications that involve bitstream adaptations to variable bandwidth conditions, is not suitable since there is no dimension of scalability in the bitstream except temporal scalability (since it is a part of H.264/AVC). A scalable multi view coding (SMVC) technique has been reported recently, which gives the advantage of scalability of SVC and the compression efficiency of MVC. Hence, SMVC can be used as an encoding technique of choice as long as it can be encoded in real-time to suit broadcast applications.

5.2.2 3D Streaming System

Figure 43 shows the end-to-end architecture of an SVC–based 3D content streaming system. The left-eye and right-eye views are encoded independently using an SVC encoder. It is also possible to use a scalable version of MVC for encoding the views as long as it can be encoded in real-time. A post-encoding operation computes the quality metadata information for each view. This step is needed as a pre-requisite for performing a rate-distortion optimal extraction of each view at a network node. Both the left and right-eye views are then simulcasted to an intermediate network node where RD optimal bitstream extraction takes place according to the available bandwidth in the channel between the node and the end-user. The extraction algorithm used for each view is the one developed in the previous chapters for the extraction of SVC encoded streams in 2D media streaming. The extracted bitstreams (left and right eye views) finally reach the decoder, where they are decoded and displayed in 3D by a 3D-capable TV or a monitor (either active shutter or passively polarized).

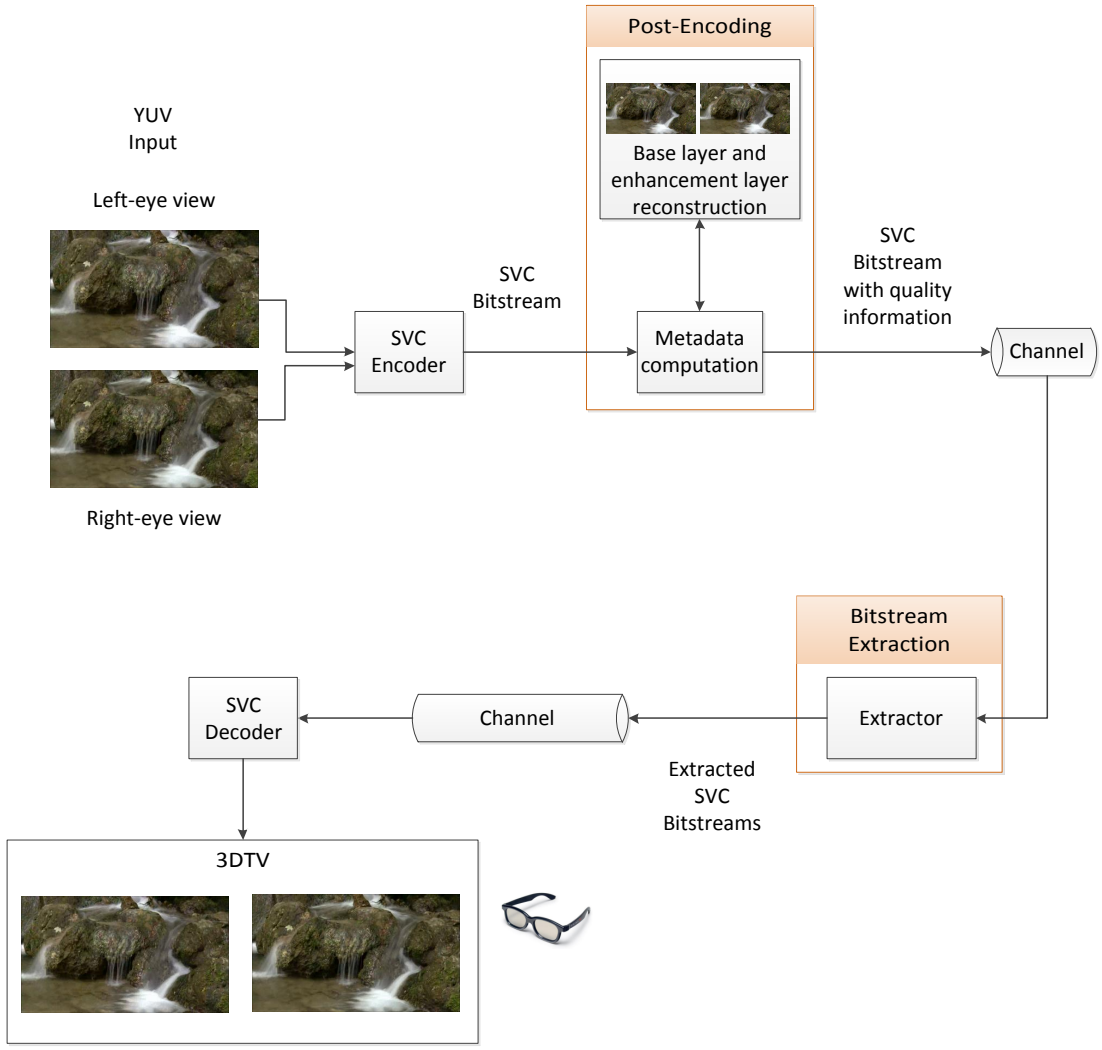


Figure 43: SVC-based 3D content streaming system – End-to-end block diagram.

5.2.3 Proposed Algorithm

The proposed architecture in Figure 43 involves the extraction of the left and right eye views so that they can be transmitted within the available bandwidth in the channel. The main goal of the extraction process is to enhance the end-user QoE under the current bandwidth conditions. For a rate-distortion optimized extraction of each stream, the available bandwidth (B) in the channel is divided by two and each bitstream is allocated a bit budget of $B/2$. With this allocation, the extraction can proceed independently for each stream in a manner identical to the extraction algorithm developed for 2D media streaming in the previous chapters. This optimal extraction of individual views produces optimal quality for left and right eye views independently. However, the end user sees both the views together with an added dimension of depth. The human brain puts together the two views and perceives depth information. The QoE here is dependent on overall perceived 3D quality, which cannot be measured by simple full reference rate-distortion metrics like mean square error for each view. Hence, the role of human perception must be factored into the extraction of the left and right-eye view streams.

5.2.3.1 Stereoscopic suppression effect

The theory of human stereo perception states that humans can perceive an overall higher quality 3D as long as one of the views is of high quality, i.e., given two views of different quality, the human brain takes the maximum of the two qualities when perceiving 3D. This effect is termed as stereoscopic suppression since the lower quality view gets suppressed and superseded by the higher quality view. For e.g., in order to perceive a scene in 3D, the brain needs the edges of the objects in the scene (the higher frequency coefficients) to be present in only one of the views. Hence, even if the other view is low-pass filtered and smoothed with no edges, the brain does not notice the smoothed view. This is very important from a compression perspective, since

now one of the views can be encoded with a finer quantization step size (resulting in a higher bitrate) and the other view can be encoded with a coarser quantization step size (resulting in a lower bitrate). It has been shown that such unequal allocation of bits would reduce the total bitrate of the left and right eye views to 1.2 times the bitrate of a single view. The reduction in resolution of one view can be spatial, temporal or quality. Authors in [96] support this theory through subjective tests conducted at various spatio-temporal resolutions on stereo video sequences. They have found that temporal subsampling between views gives unacceptable results [97] whereas spatial and quality subsampling among views are acceptable.

This theory of suppression of human stereo perception suggests that by unequally allocating bits among left and right eye views it is possible to maximize the overall perceived 3D video quality at a given bitrate or minimize the bitrate required at a given perceived 3D video quality level [23]. We use this effect to our advantage in the extraction process at the intermediate network node to maximize the overall perceived video quality for an available bandwidth of B. Instead of allocating equal bitrates ($B/2$) for extraction of each of the left and right eye views, we unequally allocate the bit budgets among the two views so that the resultant perceived quality can be maximized. Unequal allocation is implemented by simply extracting the base quality layer of all the frames in the GOP in one view and then extracting the base quality layer of the other view along with its higher MGS quality layers till the bandwidth limit is reached. This would result in one view always having a lower quality than the other view. Since the human brain takes the maximum of the two views, the overall perceived quality is the one that is represented by the higher quality view. However, there is a small percentage of population with a dominant eye effect (also known as ocular dominance). This is the effect by which the brain tends to receive most of the visual input from only one of the eyes. If one view (say left-eye view) is always of lower quality, then we run into the risk of the 3D video being perceived as very poor

quality by people who have a left dominant eye. Hence, in the extractor, we switch the unequal allocation of bit budget between the two views after every GOP. If for all the even GOPs, the left-eye view is allocated more bits, i.e., all the base quality layers and the MGS quality layers are extracted, then for all the odd numbered GOPs the right-eye view is allocated more bits. In this manner, no particular view is always of higher quality but for every GOP one of the views is of higher quality than the other. The algorithm in Figure 44 illustrates this procedure. Upon receiving a GOP of data from both the left and right eye views, the GOP # is checked for it being odd or even. For even GOPs, the base quality layer of all the frames in the GOP of the right-eye view is extracted. Then, the base quality layer of all the frames in the GOP of the left-eye view is extracted. This is followed by the extraction of the MGS quality layers one-by-one for the left-eye view till the available bandwidth limit is reached. The extraction process for the MGS quality layers is done in an RD optimal way by using the extraction algorithm developed for the streaming of regular 2D media, as shown in Figure 14. Although our 3D extraction algorithm has been described for SVC-encoded left and right-eye views, it is easily extended to cases where the views are jointly encoded using SMVC. In such cases, the base layer of the independent view is extracted and the base and MGS layers of the dependent view can be extracted to provide an overall higher perceived quality.

5.3 Experiments and Results

In this section, we discuss the subjective experiments conducted for evaluating the reconstructed 3D video quality when extracted using unequal allocation of bits among the two views as described in the previous section. The results are also compared with the subjective 3D video quality obtained when extraction is performed using equal allocation of bits among the views. Around 70% of the subjects preferred the extracted 3D video with unequal distribution of bits over equal distribution. We also

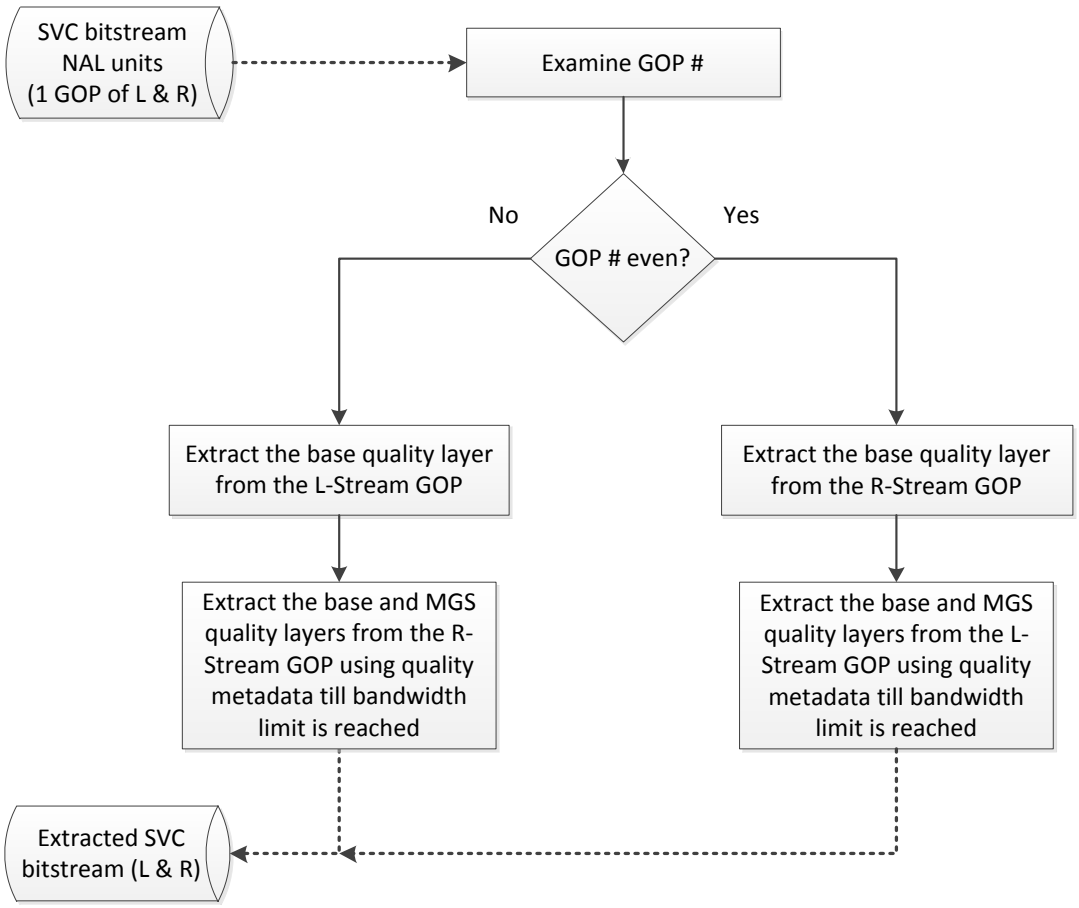


Figure 44: Proposed extraction algorithm for SVC encoded left and right-eye views.

show a few sample frames from each of the sequence that illustrate the concept of unequal bit allocation among the two views.

5.3.1 3D Content Database

Our 3D content database has a number of sequences collected from a variety of sources. Out of them, four sequences named Flower, Spider, Clownfish and Waterfall have been used in subjective experiments that evaluate the proposed extraction algorithm. Flower and Spider sequences are a part of the Macroshow sequence, obtained with permission from Gunter Peschke (`http://3d-film-peschke.de/`). Clownfish is a part of the Underwater sequence, obtained with permission from Michael Watchulonis of 3DigitalVision (`www.3DigitalVision.com`). Waterfall is a part of the Magic Forest sequence, obtained with permission from Marton Prech (`http://www.relaxvideo.hu/index_en.html`).

Each of these 4 sequences have independent left and right eye views. Sample frames from each of these sequences have been shown in Figure 45. The details of these sequences along with the various encoding parameters used are described in Table 13. The scan type is progressive in YUV 4:2:0 format. The spatial resolution is 1280×720 and the frame rate is 25 fps for all the sequences. The GOP structure uses hierarchical-B pictures, as shown in Figure 9. With a GOP size of 8 frames, the number of GOPs used is 50 for all the four sequences. All the sequences include an additional frame in the beginning, which is encoded as an IDR picture. All the sequences were encoded using the JSVM SVC encoder, which is the reference software issued by ITU–T. The quantization parameters (QP) used for encoding are also shown in Table 13. Since the GOP size used is 8, there are 4 temporal layers in all the sets. For simplicity, only one spatial layer with 6 quality layers (including the base quality layer) is used. The encoded bitrate of each layer for the left-eye views of all the four test sequences is shown in Table 14. Since the right-eye views' bitrates are very

(a) Flower

(b) Spider

(c) Clownfish

(d) Waterfall

Figure 45: Sample frames from the 3D sequence database (720p).

similar to the left-eye view bitrates, they have not been shown. From the table, we can see that the base quality layer of the left-eye views at 25 fps varies from 200 kb/s to 1000 kb/s and the maximum quality reconstruction at 25 fps varies from 2000 kb/s to 5000 kb/s.

5.3.2 Subjective Quality Evaluation

In this section, we describe the tests conducted for the subjective quality evaluation of the reconstructed 3D video. The 3D display system used is a 22-inch Samsung monitor along with Nvidia's active shutter 3D glasses. Each of the four sequences are extracted using unequal allocation of bits among the left and right-eye views using the algorithm described in Figure 44. For comparison, they are also extracted using equal allocation of bits among the two views. The extracted total bitrate is 2500 kb/s for all the sequences. This bitrate is chosen for all the sequences since it allows

Table 13: 3D sequences' characteristics and encoding parameters.

Parameter	Value
# of sequences	4
Sequence names	Flower, Spider, Clownfish and Waterfall
Spatial resolution	1280×720 (720p)
Scan type	Progressive
YUV format	4:2:0
Frame rate	25 fps
# of frames	401
Duration	16.04 s
GOP size (N)	8 frames
Sequence structure	IDR–{B3-B2-B3-B1-B3-B2-B3-P0}× 50
Base layer QP	40
MGS layer QP	30
# of Temporal layers	4 ($T = 0, 1, 2, 3$)
# of Quality layers	6 ($Q = 0, 1, 2, 3, 4, 5$)
# of Spatial layers	1 ($D = 0$)

Table 14: Bitrates (kb/s) of the SVC-encoded left-eye views of 3D test sequences (720p).

Layers (D, T, Q)	Flower	Spider	Clownfish	Waterfall
(0,0,0)	236.00	163.20	594.80	138.10
(0,1,0)	300.50	214.30	739.00	163.30
(0,2,0)	363.80	269.80	865.20	182.60
(0,3,0)	423.70	329.20	953.40	197.20
(0,0,1)	690.80	756.50	1654.00	1160.30
(0,0,2)	881.80	895.20	2253.00	1587.00
(0,0,3)	980.40	937.70	2606.00	1737.00
(0,0,4)	1023.40	952.80	2832.00	1770.00
(0,0,5)	1040.60	960.00	2916.00	1776.00
(0,1,1)	883.40	949.00	2124.00	1300.50
(0,1,2)	1101.70	1109.20	2834.00	1746.00
(0,1,3)	1213.70	1160.80	3244.00	1905.00
(0,1,4)	1264.80	1182.70	3505.00	1947.00
(0,1,5)	1288.70	1196.40	3602.00	1960.00
(0,2,1)	1070.60	1163.70	2594.00	1422.10
(0,2,2)	1313.80	1347.80	3396.00	1888.00
(0,2,3)	1443.10	1412.70	3858.00	2065.00
(0,2,4)	1508.90	1446.50	4153.00	2123.00
(0,2,5)	1546.80	1472.10	4272.00	2152.00
(0,3,1)	1260.30	1401.60	2989.00	1513.40
(0,3,2)	1535.10	1614.20	3869.00	2005.00
(0,3,3)	1692.00	1699.00	4383.00	2205.00
(0,3,4)	1785.00	1751.00	4720.00	2286.00
(0,3,5)	1849.00	1795.00	4874.00	2338.00

the extraction of the base quality layers of both the views and extra MGS quality layers of one of the views depending on the remaining bandwidth. For extraction using equal bitrate allocation among the views, a bitrate of 1250 kb/s per view is allotted and extraction is performed for each view in an RD optimal manner using the extraction algorithms proposed in the previous chapters for 2D media . For extraction using unequal bitrate allocation using our technique, the allotted bitrate for each view is not fixed. The base quality layers of one view are first extracted and then the base quality layers along with the MGS quality layers of the other view are extracted till the available bandwidth limit is reached. For the MGS layer extraction, the RD optimal technique proposed in the previous chapters for 2D media is used. For adjacent GOPs, the extraction order is switched among the two views.

Our subjects consisted of people of various age groups and various levels of expertise with video and 3D technology. There were a few 'golden-eye' video experts as well. The subjects were first trained to the 3D environment by viewing some artifact-free, high bitrate and some very low-bitrate 3D clips. This got them used to the range of artifact visibility and how does a good clip and a bad clip look in 3D. After training, they were shown two versions A and B of the same sequence. Version A was extracted using unequal allocation of bits using our algorithm while version B was extracted using equal allocation of bits. The order of A and B was mixed and the subjects were not informed of the order or any other details of the sequence. The subjective responses are shown in Table 15.

The table shows that majority of the subjects prefer unequal bitrate allocation based extraction rather than equal bitrate allocation. The reasons given by the subjects for this preference are more sharp edges and details and better overall experience. This agrees with the theory that humans perceive details from the maximum quality view. In unequal bitrate allocation, one of the views has much higher quality (switched for every GOP) when compared to the other view. Hence, all the high

Table 15: Subjective test results for perceptual video quality of 3D sequences extracted at 2500 kb/s using unequal (A) and equal (B) allocation of bits among the two views.

Sequence (2500 kb/s)	# of subjects	No preference	Prefer A	Prefer B
Flower	12	1	9	2
Spider	15	3	10	2
Clownfish	26	4	20	2
Waterfall	10	2	7	1

details are perceived from this view. For equal bitrate allocation, both views have similar quality but the quality of each view is less when compared to the higher quality view of the unequal bitrate allocation. Since the brain uses the maximum quality view for overall perception, extraction based on unequal allocation of bits wins. Averaged over all the sequences, around 70 % of the subjects prefer unequal bit allocation based extraction than equal bit allocation based extraction due to better perceived overall 3D quality.

5.3.3 Sample Frames

In this section, we show a few sample frames that show the effect of unequal bitrate distribution of bits among the left and right eye views. In Figure 46, we show a frame from the Flower sequence extracted at 2500 kb/s (total for both views) using our technique where the left-eye view is extracted at the maximum quality layer ($Q = 5$) and the right-eye view is extracted at the base quality layer ($Q = 0$). The blockiness artifacts are visible in the right-eye view in the petals and the stamen whereas the left-eye view is sharp and clear. Similarly, Figure 47 shows a frame from the Waterfall sequence extracted at 2500 kb/s. As before, the left-eye view frame is extracted at maximum quality and the right-eye view frame is extracted at minimum quality. As it can be seen from the figure, the right-eye view is smooth whereas the left-eye view

is detailed and sharp. Particularly, spots on the tree trunk, green foliage on the sides and the rocks behind the water are clearly visible in the left-eye view than in the right-eye view. Finally, Figure 48 shows a frame from the Spider sequence also extracted at 2500 kb/s. The left-eye view at maximum quality has zero artifacts where as the right-eye view at base layer quality has visible blockiness along the homogeneous background areas. As subjective tests have shown, such artifacts in one view get hidden in the overall 3D view as long as the other view is of good quality and free from visible artifacts. This is due to the human brain's nature of perceiving overall quality from the highest quality view.

5.4 Summary

In this chapter, we investigated the problem of streaming 3D media content over constrained channels where the available bandwidth varies with time. Current 3DTV technologies include frame-compatible broadcasts where the two views are downsampled and squeezed into a single frame. Though it does not place any additional requirements on network resources when compared with 2D media streaming, it leads to loss of quality by a factor of two. On the other hand, full-resolution stereo gives the best 3D quality since each view is sent at a complete resolution. However, it requires twice the bandwidth needed for regular 2D media streaming. Moreover, streaming of 3D content to a variety of end users in heterogeneous network environments requires the adaptation of each of the left and right-eye views according to the current available bandwidth in the channel. Extraction must be performed in a manner that optimizes the overall perceived quality. Hence, one of the main challenges in 3D content streaming is how to reduce the required bandwidth for full-resolution stereo while maintaining the same perceived overall 3D quality and how to maximize the perceived video quality for a given available bandwidth in the channel.

(a) Left-eye view: Maximum quality layer extraction ($Q = 5$)

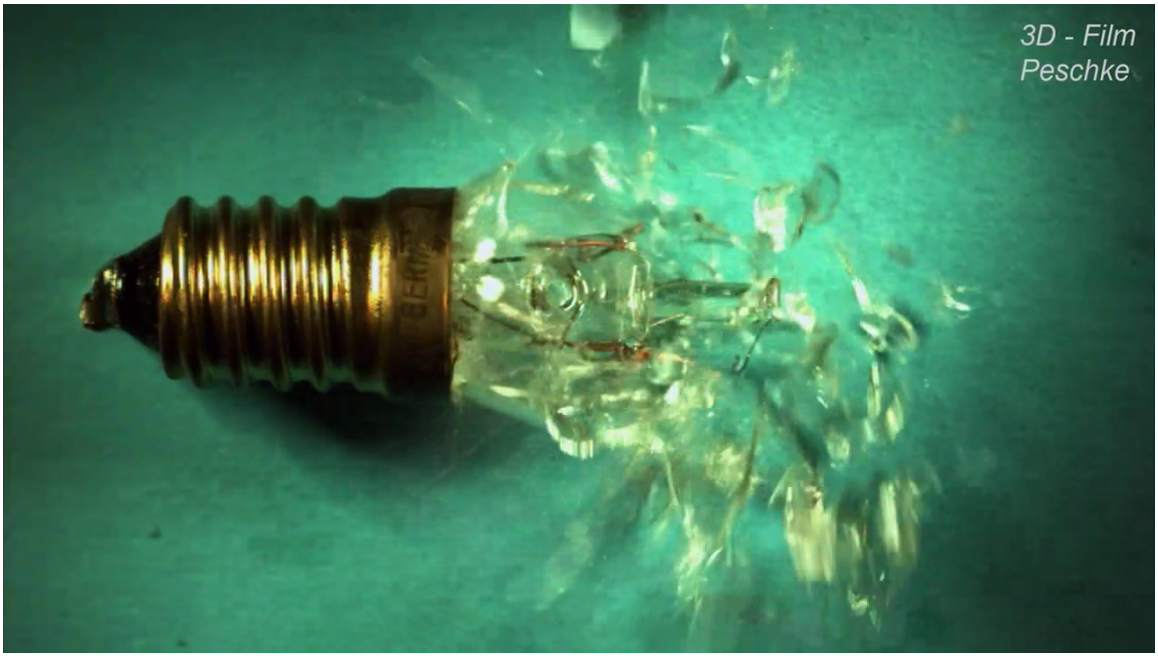
(b) Right-eye view: Base quality layer extraction ($Q = 0$)

Figure 46: Frame # 141 of Flower sequence extracted at 2500 kb/s.

(a) Left-eye view: Maximum quality layer extraction ($Q = 5$)

(b) Right-eye view: Base quality layer extraction ($Q = 0$)

Figure 47: Frame # 25 of Waterfall sequence extracted at 2500 kb/s.



(a) Left-eye view: Maximum quality layer extraction ($Q = 5$)

(b) Right-eye view: Base quality layer extraction ($Q = 0$)

Figure 48: Frame # 228 of Spider sequence extracted at 2500 kb/s.

The solution to these challenges lies in exploiting the nature of stereoscopic perception by the human brain. Given two views of different qualities that have been composed into a 3D video, the brain takes the higher of the two qualities when perceiving the overall 3D quality. The lower quality view is suppressed by the higher quality view. This is the key behind the idea of unequally allocating bit budgets while extracting the left and right eye views to maximize the overall perceived quality. As a result of such allocation, one view tends to be always at a higher quality than the other and hence, it has the potential of reducing the perceived 3D quality among people with a dominant-eye effect. Hence, we alternate the unequal bit allocation strategy between the left and the right eye views, once every GOP. This leads to a quality variation among the views. If the left-eye view has a better quality for the current GOP, then the next GOP has the right-eye view with better quality. Using SVC, we demonstrate this technique. For one view, we extract only the base quality layer and for the other view we extract the base and higher MGS quality layers as long as the available bandwidth would support it. For MGS layer extraction, RD optimal extraction techniques developed for one-way streaming of 2D content is used. For comparison purposes, extraction is also performed using equal bit allocation among the views.

Subjective evaluation of the video quality extracted with unequal and equal bit allocation among the two views confirms that unequal bit allocation has a higher overall perceived quality than equal bit allocation. On an average, 70% of the subjects prefer unequal bit allocation based extraction. Hence, overall video quality can be maximized by unequally allocating the available bandwidth for extraction among the two views. Given that the video quality is maximized by unequal allocation of bits among views for a given bitrate, it is also possible due to the same effect, that for a given video quality the required bitrate can be reduced by unequally allocating the bits among the views. This can be used in encoding of full resolution stereo 3D

videos where one view can be encoded at a higher bitrate and the other view at a lower bitrate. This results in bandwidth savings without compromise in perceived 3D video quality.

CHAPTER VI

CONCLUSIONS AND FUTURE RESEARCH

Applications such as media streaming and conferencing are the two most common forms of multimedia communication. Unprecedented growth of the Internet and reduced price of end-user devices have led to an enormous growth in these video-based communication services. To provide a high quality of experience (QoE) for the end-users, these applications depend on the underlying network characteristics since their performance is heavily degraded by network impairments such as packet loss, delay, jitter, non-availability of bandwidth, etc. Networks with zero impairments is difficult to achieve in a best-effort network like that of the Internet. Hence, the application must adapt itself to the changes in network so that it can optimize the overall QoE even during poor channel conditions. Moreover, client heterogeneity adds to the complexity of the performance of these applications. Client devices including mobile phones, PDAs, netbooks, laptops, workstations, IPTVs, etc., vary widely in their operating environment, computing power and display capabilities. They connect via heterogeneous access networks like residential broadband connections (DSL and cable), WiMAX, 3G, university campus and corporate networks. To deliver a high quality of experience (QoE) to such a variety of clients (or participants in case of a video conferencing session), it is necessary for the video content to adapt its bitrate to the changes in bandwidth and client limitations. This will help in achieving a graceful degradation when network conditions deteriorate. Content adaptation must be done at a fine granularity to ensure the best video quality possible. It should be scalable to serve a large number of clients in real time and the reaction speed should be high to enable adaptations to quick bandwidth changes. The problem is more interesting

when streaming 3D content, which requires twice the bandwidth since two bitstreams are transmitted (one for each eye) to each client and the added dimension of depth perception poses special challenges.

This thesis investigated the problem of video content adaptation to varying network resources and client limitations using the scalable video communications approach. Adaptation of scalable video in applications including video streaming, conferencing and 3DTV formed the core of the thesis. Each of these applications differ in a number of ways in terms of network requirements and the end-user expectation of QoE. For e.g., video conferencing is tightly constrained by end-to-end delay and jitter constraints along with real-time encoding. The user expectation from a video conferencing application is the ability to converse seamlessly. Streaming techniques on the other hand, do not have jitter requirements but the QoE expectations from the user is very high in terms of spatial quality, frame rate, etc. When it comes to 3D streaming, the QoE depends heavily on the perceived depth than on the quality of the individual views that make up the 3D video. Hence, we focussed on each application individually and formulated the problem of content adaptation to varying channel conditions. Solutions in terms of extraction algorithms that optimized the reconstructed video quality for a given bitrate were proposed for each application. For 3D, the objective of extraction was to optimize the perceived overall 3D video quality rather than optimizing the video quality of the individual views.

For the application of SVC-based streaming, a rate distortion optimal extraction strategy has been proposed that extracts the most important layers from the bitstream in terms of their contribution to the reconstructed video quality. The algorithm computes the quality contribution of the lowest and the highest quality layers for each frame in the bitstream as a post-encoding process. It estimates the quality contributions of the in between MGS layers and hence limits the number of

decodings. The extraction process at an intermediate network node involves extracting those layers that maximize the reduction in distortion. When compared to the current state-of-the-art techniques, our algorithm achieves a quality gain of about 1.5 dB over JSVM–Basic and a quality gain of about 0.5 dB over JSVM–QL. The maximum gain is about 4.0 dB when compared to JSVM–Basic and about 1.5 dB when compared to JSVM–QL. The time required for computing the metadata information during the post-encoding phase is 73% lesser for the proposed technique when compared with JSVM–QL. This huge reduction in metadata computation time along with the improvements in video quality make our technique a more preferred candidate than JSVM–QL and JSVM–Basic for use in real-time streaming applications. These results demonstrate the superiority of the proposed technique in delivering better video quality for a given bitrate while performing lesser number of computations for evaluating each layer's RD importance.

For SVC-based video conferencing, three extraction techniques have been proposed. Frame-by-frame extraction uses a decision window of a single frame. It ensures a jitter-free transmission and hence, it does not require a jitter-compensation buffer at the receiver. However, the reconstructed video quality of this technique is poor. The second extraction technique is paired-frame extraction. It makes extraction decisions over a window of two frames. It achieves better video quality than frame-by-frame extraction but requires a jitter compensation buffer, equal to a duration of a single frame, at the receiver. The extraction decisions are content independent and hence, are not RD optimal. The third extraction technique is paired frame extraction using quality metadata information. It makes RD optimal, content dependent extraction decisions over a window of two frames. Layers are extracted based on their contribution towards reconstructed video quality. The quality metadata information required for extraction is computed during a post-encoding operation. Allowing a quality layer

in the first frame of the pair (temporally more important) to consume extra bits originally allocated for the second frame (temporally less important) is done only when that quality layer reduces the distortion more than the competing quality layer from the second frame. Experiments show that our technique achieves a maximum quality increase of about 0.2 dB when compared to regular paired-frame extraction and a increase of about 1.3 dB when compared to frame-by-frame extraction.

The main challenge in SVC-based 3D media streaming is how to perform an optimal extraction of the left and right eye views such that it maximizes the overall perceived 3D video quality. Using the human brain's nature of stereoscopic perception, an extraction strategy has been proposed which involves extracting one view at the base quality layer and the other view at the maximum quality layer that the current available bandwidth would allow. Given two views of unequal quality, the brain always perceives the overall 3D quality from the higher quality view, i.e., the lower quality view is suppressed by the brain in favor of the better view. To take ocular dominance into account, we switch the unequal layer extraction strategy between the views once after every GOP. Subjective evaluation has shown that 70% of the subjects prefer the extracted video with unequal allocation of bits to left and right eye views over equal allocation of bits among the views.

The key contributions of this thesis can be summarized as the proposal of bitstream extraction algorithms for scalable video coding (SVC) based streaming, conferencing and 3DTV. Based on a combination of metadata computations and prediction mechanisms, these algorithms evaluate the quality contribution of each layer in the SVC bitstream and make extraction decisions that are aimed at maximizing video quality while operating within the available bandwidth resources. These techniques have been applied in two-way interaction and one-way streaming of 2D and 3D content. Depending on the delay tolerance of these applications, rate-distortion optimized extraction algorithms have been proposed. For conferencing applications, the

extraction decisions are made over single frames and frame pairs due to tight end-to-end delay constraints. The proposed extraction algorithms for 3D content streaming maximize the overall perceived 3D quality based on human stereoscopic perception. When compared to current extraction methods, the new algorithms offer better video quality at a given bitrate while performing lesser number of metadata computations in the post-encoding phase. The solutions proposed for each application achieve the recurring goal of maintaining the best possible level of end-user quality of multimedia experience in spite of network impairments.

This research work has a number of possible extensions. More complex optimization mechanisms could be adopted for extracting the various quality layers from the bitstream. Additional prediction mechanisms can help reduce the number of computations performed during the post-encoding operation. This problem becomes more interesting when multiple spatial layers are involved. A joint optimization of the various MGS layers across the different spatial resolutions can be performed. For video conferencing, the current technique can be extended to include QoS priorities, which is common in enterprise networks. The domain of 3DTV is new and open to a number of research challenges. Perceived 3D quality is also a function of the display type (active shutter, passive polarization or glasses-free 3D). It would be interesting to observe the effect of spatial and temporal subsampling among the two views and the response of the subjects to such a change in resolution between the views. Other content-dependent extraction strategies that also reflect unequal bitrate allocation could be an excellent extension. Another area to investigate would be the effect of viewing longer duration 3D sequences, such as movies, on human stereo perception.

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] S. Wenger, "H.264/AVC over IP," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 645–656, Jul. 2003.

[3] T. Stockhammer, M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 657–673, Jul. 2003.

[4] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *Circuits and Systems Magazine, IEEE*, vol. 4, no. 1, pp. 7–28, 2004.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[6] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1194–1203, Sep. 2007.

[7] H. Sun, A. Vetro, and J. Xin, "An overview of scalable video streaming," *Wireless Communications and Mobile Computing*, vol. 7, no. 2, pp. 159–72, Feb. 2007.

[8] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-time system for adaptive video streaming based on SVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1227–1237, Sep. 2007.

[9] A. Eleftheriadis, M. Civanlar, and O. Shapiro, "Multipoint videoconferencing with scalable video coding," *Journal of Zhejiang University - Science A*, vol. 7, no. 5, pp. 696–705, May 2006. [Online]. Available: http://dx.doi.org/10.1631/jzus.2006.A0696

[10] T. Wiegand, L. Noblet, and F. Rovati, "Scalable video coding for IPTV services," *Broadcasting, IEEE Transactions on*, vol. 55, no. 2, pp. 527–538, Jun. 2009.

[11] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet protocol television (IPTV): The killer application for the next-generation internet," *Communications Magazine, IEEE*, vol. 45, no. 11, pp. 126 –134, Nov. 2007.

[12] S. Park and S.-H. Jeong, "Mobile IPTV: Approaches, challenges, standards, and qos support," *Internet Computing, IEEE*, vol. 13, no. 3, pp. 23 –31, May-Jun. 2009.

[13] T. Schierl, T. Stockhammer, and T. Wiegand, "Mobile video transmission using scalable video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1204–1217, Sep. 2007.

[14] M. van der Schaar and P. A. Chou, *Multimedia over IP and Wireless Networks : Compression, Networking and Systems*, M. van der Schaar and P. A. Chou, Eds. Academic Press, 2007.

[15] *ITU-T Recommendation G.114: One–way transmission time*, ITU Telecommunication Standardization Sector (ITU-T) Std., 2003.

[16] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*. Cisco Press, Nov. 2004.

[17] *Enterprise QoS Solution Reference Network Design Guide*, Cisco Press, Nov. 2005.

[18] "Web server vs. streaming server," http://www.microsoft.com/windows/windowsmedia/compare/WebServVStreamServ.aspx.

[19] W. R. Stevens, *TCP/IP Illustrated, Volume 1 : The Protocols*. Addison-Wesley, 1994.

[20] C. Perkins, *RTP : Audio and Video for the Internet*. Addison-Wesley, 2003.

[21] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," RFC 2326 (Proposed Standard), Internet Engineering Task Force, Apr. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2326.txt

[22] A. Vetro, "Representation and coding formats for stereo and multiview video," in *Intelligent Multimedia Communication: Techniques and Applications*, ser. Studies in Computational Intelligence, C. Chen, Z. Li, and S. Lian, Eds. Springer, 2010, vol. 280, pp. 51–73.

[23] A. Tekalp, E. Kurutepe, and M. Civanlar, "3DTV over IP," *Signal Processing Magazine, IEEE*, vol. 24, no. 6, pp. 77–87, Nov. 2007.

[24] D. Wu, Y. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha, "Streaming video over the Internet: approaches and directions," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 3, pp. 282–300, Mar. 2001.

[25] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP : A transport protocol for real-time applications," RFC 3550 (Standard), Internet Engineering Task Force, Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[26] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP : Session initiation protocol," RFC 3261 (Proposed Standard), Internet Engineering Task Force, Jun. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[27] S. Wenger, Y.-K. Wang, and T. Schierl, "Transport and signaling of SVC in IP networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1164–1173, Sep. 2007.

[28] T. Schierl, K. Gruneberg, and T. Wiegand, "Scalable video coding over RTP and MPEG-2 transport stream in broadcast and IPTV channels," *Wireless Communications, IEEE*, vol. 16, no. 5, pp. 64–71, Oct. 2009.

[29] D. Renzi, P. Amon, and S. Battista, "Video content adaptation based on SVC and associated RTP packet loss detection and signaling," in *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS '08. Ninth International Workshop on*, May 2008, pp. 97–100.

[30] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP) – version 1 functional specification," RFC 2205 (Proposed Standard), Internet Engineering Task Force, Sep. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2205.txt

[31] M. Handley, C. Perkins, and E. Whelan, "Session announcement protocol," RFC 2974 (Experimental), Internet Engineering Task Force, Oct. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2974.txt

[32] M. Handley, V. Jacobson, and C. Perkins, "SDP : Session description protocol," RFC 4566 (Proposed Standard), Internet Engineering Task Force, Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4566.txt

[33] A. M. Bock, *Video Compression Systems - From first principles to concatenated codecs*. IET, 2009.

[34] *ITU-T Recommendation J.144: Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*, ITU Telecommunication Standardization Sector (ITU-T) Std., 2001.

[35] N. Suresh and N. Jayant, "Mean time between failures: A subjectively meaningful video quality metric," in *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, May 2006, pp. 941–944.

[36] N. Suresh, N. Jayant, and O. Yang, "Mean time between failures: A subjectively meaningful quality metric for consumer video," in *Invited Talk, Proceedings of the Second International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2006.

[37] N. Suresh, O. Yang, and N. Jayant, "AVQ: a zero-reference metric for automatic measurement of the quality of visual communications," in *Invited Talk, Proceedings of the Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2007.

[38] N. Suresh, P. Mane, and N. Jayant, "Real-time prototype of a zero-reference video quality algorithm," in *International Conference on Consumer Electronics*, Jan. 2008.

[39] N. Jayant, N. Suresh, P. Mane, and R. Palaniappan, "Quality of user experience in next generation television," in *Broadband Multimedia Systems and Broadcasting, IEEE International Symposium on*, May 2009.

[40] R. Palaniappan, N. Suresh, and N. Jayant, "Objective measurement of transcoded video quality in mobile applications," in *a World of Wireless, Mobile, and Multimedia Networks, IEEE International Symposium on*, Jun. 2008.

[41] N. Suresh, R. Palaniappan, P. Mane, and N. Jayant, "Testing of a no reference VQ metric : monitoring quality and detecting visible artifacts," in *Video Processing and Quality Metrics for Consumer Electronics, Fourth International Workshop on*, 2009.

[42] R. Palaniappan, N. Suresh, and N. Jayant, "Taxonomy of video artifacts," Georgia Tech Broadband Institute, Tech. Rep., Jul. 2008.

[43] H. Schwarz and M. Wien, "The scalable video coding extension of the H.264/AVC standard [standards in a nutshell]," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 135–141, Mar. 2008.

[44] T. Oelbaum, H. Schwarz, M. Wien, and T. Wiegand, "SVC overview and performance evaluation," in *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 7073, San Diego, CA, United states, 2008, pp. 1–12. [Online]. Available: http://dx.doi.org/10.1117/12.797351

[45] *ITU-T Recommendation H.264 (03/2009)*, ITU Telecommunication Standardization Sector (ITU-T) Std., 2009.

[46] R. Palaniappan and N. Jayant, "N-way video communication over enterprise networks based on adaptive bit stream extraction in scalable video coding," in *Image and Video Technology, Fourth Pacific Rim Symposium on*, 2010.

[47] C. Segall and G. Sullivan, "Spatial scalability within the H.264/AVC scalable video coding extension," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1121–1135, Sep. 2007.

[48] Y.-K. Wang, M. Hannuksela, S. Pateux, A. Eleftheriadis, and S. Wenger, "System and transport interface of SVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1149–1163, Sep. 2007.

[49] G. Conklin, G. Greenbaum, K. Lillevold, A. Lippman, and Y. Reznik, "Video coding for streaming media delivery on the Internet," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 3, pp. 269–281, Mar. 2001.

[50] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.

[51] G. Keesman, R. Hellinghuizen, I. F. Hoeksema, and I. G. Heideman, "Transcoding of MPEG bitstreams," *Signal processing: image communication*, vol. 8, no. 6, pp. 480–500, 1996. [Online]. Available: http://doc.utwente.nl/14849/

[52] V. Goyal, "Multiple description coding: compression meets the network," *Signal Processing Magazine, IEEE*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[53] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005.

[54] A. Ortega and H. Wang, "Mechanisms for adapting compressed multimedia to varying bandwidth conditions," in *Multimedia over IP and Wireless Networks : Compression, Networking and Systems*, P. A. Chou and M. van der Schaar, Eds. Academic Press, 2007, ch. 4, pp. 81–116.

[55] H. Kalva, "Issues in H.264/MPEG-2 video transcoding," in *First IEEE Consumer Communications and Networking Conference*, Jan. 2004, pp. 657–659.

[56] J. Apostolopoulos, T. Wong, W. tian Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, 2002, pp. 1736–1745.

[57] J. Apostolopoulos, M. Trott, and W.-T. Tan, "Path diversity for media streaming," in *Multimedia over IP and Wireless Networks : Compression, Networking and Systems*, P. A. Chou and M. van der Schaar, Eds. Academic Press, 2007, ch. 17, pp. 559–590.

[58] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[59] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. Wiley, 2010.

[60] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1186–1193, Sep. 2007.

[61] E. Maani and A. Katsaggelos, "Optimized bit extraction using distortion modeling in the scalable extension of H.264/AVC," *Image Processing, IEEE Transactions on*, vol. 18, no. 9, pp. 2022 –2029, Sep. 2009.

[62] L. Wei, G. Sen, C. Xu, and Z. Jihong, "SVC bitstream extraction based on the importance of MGS slice," in *Industrial and Information Systems (IIS), Second International Conference on*, vol. 1, Jul. 2010, pp. 148 –151.

[63] J. Lu, S. Xiao, and C. Wu, "Optimized state-distortion extraction for scalable extension of H.264/AVC," in *Intelligent Information Hiding and Multimedia Signal Processing, Sixth International Conference on*, oct. 2010, pp. 599 –602.

[64] W.-H. Peng, L.-S. Huang, J. K. Zao, J.-S. Lu, T.-W. Wang, H.-T. Huang, and L.-C. Kuo, "Rate-distortion optimized SVC bitstream extraction for heterogeneous devices: A preliminary investigation," in *Ninth IEEE Int. Symp. Multimedia Workshops*, 2007, pp. 407–412.

[65] N. Cranley, P. Perry, and L. Murphy, "Optimum adaptation trajectories for streamed multimedia," *Multimedia Systems*, vol. 10, pp. 392–401, 2005.

[66] Y. Wang, S.-F. Chang, and A. Loui, "Subjective preference of spatio-temporal rate in video adaptation using multi-dimensional scalable coding," in *Multimedia and Expo, IEEE International Conference on*, vol. 3, Jun. 2004, pp. 1719–1722.

[67] Y. Wang, M. van der Schaar, S.-F. Chang, and A. Loui, "Classification-based multidimensional adaptation prediction for scalable video coding using subjective quality evaluation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 10, pp. 1270–1279, Oct. 2005.

[68] Y. S. Kim, Y. J. Jung, T. C. Thang, and Y. M. Ro, "Bit-stream extraction to maximize perceptual quality using quality information table in SVC," in *Visual Communications and Image Processing*, J. G. Apostolopoulos and A. Said, Eds., vol. 6077, no. 1. SPIE, 2006, pp. 1–11.

[69] G. Zhai, J. Cai, W. Lin, X. Yang, W. Zhang, and M. Etoh, "Cross-dimensional perceptual quality assessment for low bit-rate videos," *Multimedia, IEEE Transactions on*, vol. 10, no. 7, pp. 1316–1324, Nov. 2008.

[70] http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm.

[71] V. Vasudevan, S. Sengupta, and J. Li, "A first look at media conferencing traffic in the global enterprise," in *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, Berlin, Germany, 2009, pp. 133–142.

[72] S.-H. Chung, Y. Won, D. Agrawal, S.-C. Hong, J. W.-K. Hong, H.-T. Ju, and K. Park, "Detection and analysis of packet loss on underutilized enterprise network links," in *End-to-End Monitoring Techniques and Services, Workshop on*, May 2005, pp. 164–176.

[73] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney, "A first look at modern enterprise traffic," in *Proceedings of the Fifth ACM SIGCOMM conference on Internet Measurement*. USENIX Association, 2005.

[74] T. Karagiannis and R. Mortier, "Address and traffic dynamics in a large enterprise network," in *Local and Metropolitan Area Networks, 16th IEEE Workshop on*, Sep. 2008, pp. 102–107.

[75] M. Li and C.-R. Chang, "A two-way available bandwidth estimation scheme for multimedia streaming networks adopting scalable video coding," in *Sarnoff Symposium, IEEE*, Mar. 2009, pp. 1–6.

[76] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 4, pp. 537–549, Aug. 2003.

[77] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," *Selected Areas in Communications, IEEE Journal on*, vol. 21, no. 6, pp. 879–894, Aug. 2003.

[78] J. Kilpi and I. Norros, "Testing the gaussian approximation of aggregate traffic," in *Proceedings of the Second ACM SIGCOMM Workshop on Internet measurment*, 2002, pp. 49–61.

[79] J. Ribas-Corbera, P. Chou, and S. Regunathan, "A generalized hypothetical reference decoder for H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 674–687, Jul. 2003.

[80] M. Schubin, "What 3D is and why it matters," in *Spring Technical Forum Proceedings*, 2010.

[81] A. Puri, R. V. Kollarits, and B. G. Haskell, "Basics of stereoscopic video, new compression results with MPEG-2 and a proposal for MPEG-4," *Signal Processing: Image Communication*, vol. 10, no. 1-3, pp. 201–234, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/B6V08-3SNVM2V-X/2/0624f754b5016c07096dc50158433b23

[82] O. Schreer, P. Kauff, and T. Sikora, *3D Video Communication: Algorithms, concepts and real-time systems in human centred communication*. Wiley, 2005.

[83] H. M. Ozaktas and L. Onural, *Three-Dimensional Television: Capture, Transmission, Display*. Springer, 2009.

[84] *ITU-T Recommendation H.262 (MPEG 2)*, ITU Telecommunication Standardization Sector (ITU-T) Std., 2000.

[85] A. Garcia-Crespo, F. Paniagua-Martin, R. Colomo-Palacios, and J. M. Gomez-Berbis, "Visual fatigue in three-dimensional subtitle projections," in *Stereoscopic 3D for Media and Entertainment, SMPTE International Conference on*, 2010.

[86] D. Howard, M. Green, R. Palaniappan, and N. Jayant, "Visibility of digital video artifacts in stereoscopic 3DTV," in *Stereoscopic 3D for Media and Entertainment, SMPTE International Conference on*, 2010.

[87] ——, "Visibility of digital video artifacts in stereoscopic 3DTV," *SMPTE Motion Imaging Journal*, vol. May/Jun., pp. 49–53, 2011.

[88] P. Larbier, "3D: How video compression technology can contribute," in *Stereoscopic 3D for Media and Entertainment, SMPTE International Conference on*, 2010.

[89] Y. Chen, Y.-K. Wang, K. Ugur, M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 786015, pp. 1–13, 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/786015

[90] D. K. Broberg and M. Francisco, "Implementation of stereoscopic 3D systems on cable," in *Spring Technical Forum Proceedings*, 2010.

[91] W. Husak, "Stereoscopic delivery of 3D content to the home," in *Spring Technical Forum Proceedings*, 2010.

[92] G. B. Akar and A. Gotchev, "MOBILE 3DTV: content delivery optimization over DVB-H system," in *Stereoscopic 3D for Media and Entertainment, SMPTE International Conference on*, 2010.

[93] P. H. Putman, "Display technologies for consumer 3D TV viewing compared and contrasted," in *Stereoscopic 3D for Media and Entertainment, SMPTE International Conference on*, 2010.

[94] P. D. Panabaker and S. S. Cho, "Quality autostereoscopic displays," in *Stereoscopic 3D for Media and Entertainment, SMPTE International Conference on*, 2010.

[95] G. Akar, A. Tekalp, C. Fehn, and M. Civanlar, "Transport methods in 3DTV – A survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1622–1630, Nov. 2007.

[96] L. Stelmach, W. J. Tam, D. Meegan, and A. Vincent, "Stereo image quality: effects of mixed spatio-temporal resolution," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 2, pp. 188 –193, Mar. 2000.

[97] N. Ozbek, A. Tekalp, and E. Tunali, "Rate allocation between views in scalable stereo video coding using an objective stereo video quality measure," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*, vol. 1, Apr. 2007, pp. I1045 –I1048.

VITA

Ramanathan Palaniappan received his Bachelor of Engineering (BE) degree from Anna university, India in 2006. In December 2008, he received his Master of Science (MSECE) degree from the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, USA, from where he is also receiving his Doctor of Philosophy (PhD) degree in December 2011, all in electrical engineering. He interned with Qualcomm Inc. during the summers of 2007 and 2009. His current research focuses on human stereoscopic perception, scalable video coding and its applications. His general research interests include video compression and communication, error control techniques, video quality evaluation including challenges in IPTV, mobile TV, 3D video and high quality telepresence.